NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

AN FVE-FAC APPROACH FOR THE WELDPOOL PROBLEM

by

Thomas E. Rogers

September, 1995

Thesis Advisor: Van Emden Henson

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE September, 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE AN FVE-FAC APPROACH FOR THE WELDPOOL PROBLEM	5. FUNDING NUMBERS
6. AUTHORS Rogers, Thomas E.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT(*maximum 200 words*)

A block of metal is subjected to a concentrated heat source resulting in a pool of molten metal surrounded by a portion of the unmelted metal. The governing system of equations is known as the weldpool problem. The weldpool problem is discretized using finite differences to discretize time derivatives and the Finite Volume Element method (FVE) to discretize spatial derivatives. Multigrid methods, known to be effective on uniform grids, make use of overlapping uniform grids of different scales to better approximate solutions to the weldpool problem. However, the solid-liquid interface requires extremely fine grid resolution and accuracy to resolve the physical behavior of the weldpool problem at this interface. Being too costly to apply a global fine domain, a non-uniform domain is developed to utilize finer resolution along the interface while still maintaining a coarser resolution on the rest of the domain. The fast adaptive composite grid method (FAC) is introduced, incorporating the concepts of multigrid to solve the weldpool problem on this non-uniform discrete domain. FVE is then applied to the conduction equation in the solid and the convection-diffusion equation in the liquid metal to develop stencil equations for use in FAC.

14. SUBJECT TERMS Finite Volume Element (FVE), Fast Adaptive Composite (FAC), Multigrid	15. NUMBER OF PAGES 92
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited

AN FVE-FAC APPROACH FOR THE WELDPOOL PROBLEM

Thomas E. Rogers
Lieutenant, United States Navy
B.S., United States Naval Academy, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 1995

Author: Thomas E Rogers

Thomas E. Rogers

Approved by:

Van Henson, Advisor

David Canright, Second Reader

Richard Franke, Chairman, Department of Mathematics

ABSTRACT

A block of metal is subjected to a concentrated heat source resulting in a pool of molten metal surrounded by a portion of the unmelted metal. The governing system of equations is known as the weldpool problem. The weldpool problem is discretized using finite differences to discretize time derivatives and the Finite Volume Element method (FVE) to discretize spatial derivatives. Multigrid methods, known to be effective on uniform grids, make use of overlapping uniform grids of different scales to better approximate solutions to the weldpool problem. However, the solid-liquid interface requires extremely fine grid resolution and accuracy to resolve the physical behavior of the weldpool problem at this interface. Being too costly to apply a global fine domain, a non-uniform domain is developed to utilize finer resolution along the interface while still maintaining a coarser resolution on the rest of the domain. The fast adaptive composite grid method (FAC) is introduced, incorporating the concepts of multigrid to solve the weldpool problem on this non-uniform discrete domain. FVE is then applied to the conduction equation in the solid and the convection-diffusion equation in the liquid metal to develop stencil equations for use in FAC.

LIST OF FIGURES

I. INTRODUCTION

Several practical materials processes, such as welding and float-zone purification, involve a pool of molten metal with a free surface, with strong temperature gradients along the surface. In certain instances, thermocapillary forces are the primary drivers of convection in the liquid metal. In cases where other forces are stronger overall, the thermocapillary forces may still be dominant at the edges of the molten pool. Strong thermocapillary convection can lead to localized intense heat transfer and high velocities where the liquid free surface contacts the solid. This localized heat transfer can modify the shape of the solid-liquid interface that bounds the molten pool [Ref. 1].

Solving problems such as this can be challenging. One question that arises immediately is how to discretize a continuum problem of this type. It is then necessary to construct a numerical process that will solve the resulting discrete system, which also poses difficulties. The purpose of this work is twofold. First, the weldpool continuum problem must be discretized and some numerical process developed to solve the discrete system. Methods for discretization will be discussed, and then incorporated into a method for finding a solution to the weldpool problem. Next, tools will be developed that result from this discussion.

Several methods for discretizing and solving the system will be discussed. One method is the finite volume element (FVE) method, which has been shown to be an effective instrument in developing discretizations, particularly for problems that require the enforcement of conservation laws. Once discretized, the system of equations must be solved. Another technique, multigrid, has enjoyed remarkable success in solving certain types of linear and

non-linear partial differential equations (PDEs) in a uniformly discretized domain.

Another aspect of the problem results from examination of the solid-liquid interface. Near this boundary, the full conduction-convection solution of the weldpool problem will require extremely fine length scales to resolve the physical behavior of the system. Multigrid is an efficient solver on a grid of uniform length scale, but it is not effective in cases that require fine grid resolution and accuracy on only a portion of the global domain. The fast adaptive composite grid method (FAC) is a natural extension of the multigrid methods applied to problems discretized on domains of varying length scales. Again, FVE serves admirably as the tool for spatial discretization.

To better explain how the above techniques can be applied to the weldpool problem, underlying ideas are first developed by applying the techniques to simple model problems. For instance, to illustrate the method of finite differences, the time derivatives of the one-dimensional diffusion equation are discretized. To illustrate the finite volume element approach, the potential flow problem in two-dimensional space is discretized. Having developed an understanding of FVE discretization, the system of equations governing a simplified weldpool problem are discretized, taking into account the axisymmetric nature of the problem.

FAC is then introduced as a method to deal with the need for extremely fine length scales along the boundary of the liquid-solid interface. FAC is an extension of multigrid concepts. Thus, multigrid is first developed for two model problems, a two-point boundary value problem describing the steady-state temperature of a long uniform rod, and the second-order PDE known as Poisson's equation. With an understanding of multigrid in hand, FAC

is developed by building on concepts from multigrid applied to a domain of varying length scale. Again, a simplified model problem is used to illustrate these techniques. In this case, the model problem is the potential flow problem in two-dimensional space. Finally, discretization of the simplified weldpool problem is developed for a domain with multiple length scales.

The discretized equations developed are for a model of the weldpool problem, while the real problem is far more complicated. This work contributes to the process of solving the general weldpool problem by providing numerical tools for a simplified version of the problem. Also, complications and difficulties in the general problem may come to light by running the FAC discretization process on the simplified problem.

II. PROBLEM STATEMENT

A semi-infinite block of metal is subjected to a concentrated heat source applied to the horizontal surface and has an inviscid nonconducting gas above the surface. The result is a pool of molten metal with a free surface surrounded by the solid portion of the unmelted block (Figure 1). The following development of the weldpool problem is a recapitulation of Canright [Ref. 1].

The surface tension of the liquid is assumed strong enough to keep the free surface flat, but with surface tension variations resulting from a linear dependence on temperature, T. The resulting thermal and flow fields are assumed to be axisymmetric and steady. The time-dependent equations are given to facilitate a numerical approach using time-like iterations to reach a steady state solution [Ref. 1]. The governing system of equations is

$$\text{solid} \quad : \quad \frac{\partial T}{\partial t} = \kappa \nabla^2 T \tag{II.1}$$

$$\text{liquid} \quad : \quad \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla \mathrm{T} = \kappa \nabla^2 \mathrm{T} \tag{II.2}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla \mathrm{p} + \nu \nabla^2 \mathbf{u} \tag{II.3}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{II.4}$$

with the conditions at the boundaries and at the solid-liquid interface given as

$$\text{solid surface } z = 0 \quad : \quad \frac{\partial T}{\partial z} = 0$$

$$\text{liquid surface } z = 0 \quad : \quad k \frac{\partial T}{\partial z} = -q(r)$$

$$v = 0$$

$$\mu \frac{\partial u}{\partial z} = -\gamma \frac{\partial T}{\partial r}$$

$$\text{axis } r = 0 \quad : \quad \frac{\partial T}{\partial r} = 0$$

$$u = 0$$

$$\frac{\partial v}{\partial r} = 0.$$

This system is governed by conservation of energy in the solid and by conservation of energy, momentum, and mass in the pool. In the system, κ is thermal diffusivity, t is time, ρ is density, p is pressure, ν is kinematic viscosity, $\mathbf{u}$ is velocity with components u and v in the r and z directions (cylindrical coordinates), k is thermal conductivity, $q(r)$ is imposed surface heat flux (large at $r = 0$ and falling off to zero at some small value of r such that $\int_0^\infty q(r) 2\pi r dr = Q$), μ is viscosity, and γ is the negative of the derivative of the surface tension with respect to temperature (γ is assumed to be constant and positive).

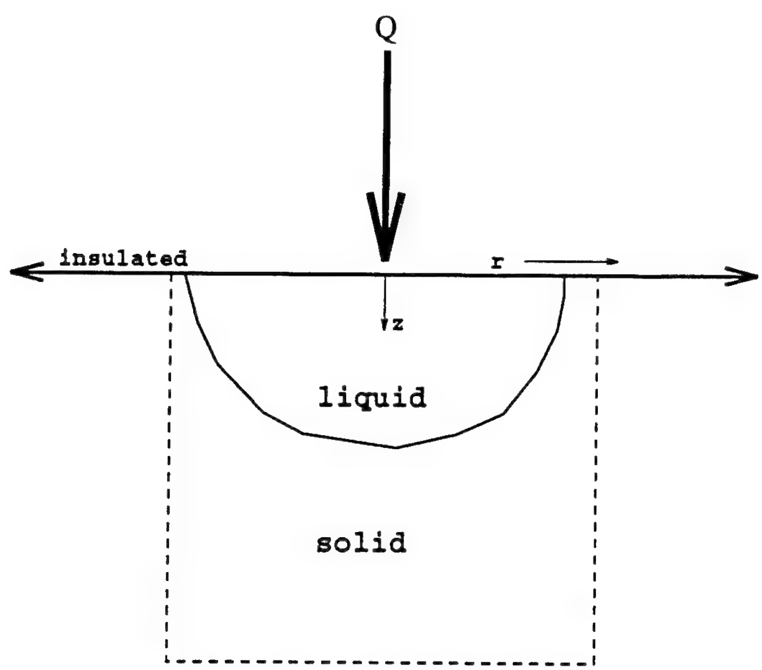

Figure 1. *A block of pure metal is subjected to a concentrated surface heating, Q, that results in a molten pool.*

The equations can be nondimensionalized. The main dimensionless parameters are the Marangoni number Ma and the Reynolds number Re. It is also convenient to eliminate the pressure by adopting a stream function/vorticity formulation for the flow [Ref. 1]:

$$\begin{aligned} Re\left(\frac{\partial \omega}{\partial t} - \nabla \times (\mathbf{u} \times \omega)\right) &= -\nabla \times \nabla \times \omega \\ \omega &= \nabla \times \nabla \times \left(\frac{\Psi}{r}\hat{\theta}\right) \\ \mathbf{u} = \nabla \times \left(\frac{\Psi}{\mathrm{r}}\right)\hat{\theta} &= -\frac{1}{r}\frac{\partial \Psi}{\partial z}\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial \Psi}{\partial r}\hat{\mathbf{z}}, \end{aligned}$$

where Ψ is the axisymmetric stream function and ω is the vorticity vector (having only one component, in the $\hat{\theta}$ direction). The flow boundary conditions are

$$\begin{aligned} \text{liquid surface } z = 0 \quad &: \quad \Psi = 0 \\ & \quad\;\; \omega = \frac{\partial T}{\partial r} \\ \text{axis } r = 0 \quad &: \quad \Psi = 0 \\ & \quad\;\; \omega = \mathbf{0} \\ \text{liquid/solid boundary } r = f(z,t) \quad &: \quad \Psi = \frac{\partial \Psi}{\partial r} = \frac{\partial \Psi}{\partial z} = 0 \end{aligned}$$

All of this results in the following equations:

$$\text{solid} \quad : \quad \frac{\partial T}{\partial t} = Ma^{-1}\nabla^2 T \tag{II.5}$$

$$\text{liquid} \quad : \quad \frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}\mathrm{T}) = \mathrm{Ma}^{-1}\nabla^2\mathrm{T} \tag{II.6}$$

$$\frac{\partial \omega}{\partial t} - \nabla \times (\mathbf{u} \times \omega) = -\mathrm{Re}^{-1}\nabla \times \nabla \times \omega \tag{II.7}$$

$$\omega = \nabla \times \nabla \times \left(\frac{\Psi}{r}\right)\hat{\theta} \tag{II.8}$$

$$\text{where} \qquad \mathbf{u} = \nabla \times \left(\frac{\Psi}{\mathrm{r}}\right)\hat{\theta} = -\frac{1}{\mathrm{r}}\frac{\partial \Psi}{\partial \mathrm{z}}\hat{\mathbf{r}} + \frac{1}{\mathrm{r}}\frac{\partial \Psi}{\partial \mathrm{r}}\hat{\mathbf{z}} \tag{II.9}$$

with the conditions at the boundaries and at the solid-liquid boundary given as

$$
\begin{aligned}
\text{solid surface } z = 0 \quad &: \quad \frac{\partial T}{\partial z} = 0 \\
\text{liquid surface } z = 0 \quad &: \quad \frac{\partial T}{\partial z} = -q(r) \\
&\quad\ \Psi = 0 \\
&\quad\ \omega = \frac{\partial T}{\partial r}\hat{\theta} \\
\text{axis } r = 0 \quad &: \quad \frac{\partial T}{\partial r} = 0 \\
&\quad\ \Psi = 0 \\
&\quad\ \omega = \mathbf{0}.
\end{aligned}
$$

Refer to Canright [Ref. 1] for a more in-depth development of the weldpool problem.

This system of equations and systems like it present interesting challenges for constructing numerical solutions. The goal is to transform this continuum problem into a discrete one and to develop a numerical process to solve it. The Finite Volume Element Method (FVE) has proven to be a useful tool in discretizing systems involving the enforcement of conservation laws. Multi-level techniques, such as multigrid, have been effective in streamlining the solution of partial differential equations (PDE's). The Fast Adaptive Composite method (FAC) is designed to efficiently discretize and find solutions for PDE's. FAC methods are characterized by their use of a composite grid, the union of various uniform grids. FAC methods require discretization of the PDE on both the individual levels and the composite grid. FVE is utilized for this purpose.

III. DISCRETIZATION

Numerically solving the conduction problem

$$\begin{aligned} \frac{\partial T}{\partial t} &= \kappa \nabla^2 T \\ \text{surface} \quad z = 0 &: \quad \kappa \frac{\partial T}{\partial z} = -\delta(r) \\ \text{axis} \quad r = 0 &: \quad \frac{\partial T}{\partial r} = 0 \\ \text{far boundaries} \quad r, z = 1 &: \quad T = 0 \end{aligned} \tag{III.1}$$

requires discretization of both the space and time derivatives, where the value of the unknown function T is determined from a set of equations which approximates equation (III.1). In the following, finite differences are used to discretize the time derivatives, while the Finite Volume Element (FVE) method is used to discretize the spatial derivatives.

A. FINITE DIFFERENCES AND CRANK-NICOLSON

Discretizing in both time and space can be complicated, and it may be worthwhile to develop the method on a simple one-dimensional equation. To illustrate the discretization of the time derivatives, consider the one-dimensional diffusion equation,

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}. \tag{III.2}$$

Assuming the function T and its derivatives are single-valued, finite and continuous functions of x and t, then Taylor's theorem can be used to approximate each term of (III.2). Holding t constant (i.e., treating T as a function of x only) and applying Taylor's theorem gives

$$T(x+\triangle x) = T(x) + \triangle x\frac{\partial T}{\partial x} + \frac{1}{2}\triangle x^2\frac{\partial^2 T}{\partial x^2} + \frac{1}{6}\triangle x^3\frac{\partial^3 T}{\partial x^3} + \ldots \tag{III.3}$$

and

$$T(x-\triangle x) = T(x) - \triangle x\frac{\partial T}{\partial x} + \frac{1}{2}\triangle x^2\frac{\partial^2 T}{\partial x^2} - \frac{1}{6}\triangle x^3\frac{\partial^3 T}{\partial x^3} + \ldots. \tag{III.4}$$

Summing these expressions gives

$$T(x+\triangle x) + T(x-\triangle x) = 2T(x) + \triangle x^2\frac{\partial^2 T}{\partial t^2} + O(\triangle \mathrm{x}^4),$$

where $O(\triangle \mathrm{x}^4)$ denotes the terms containing fourth and higher powers of $\triangle x$. These terms are assumed to be negligible compared to the lower powers of $\triangle x$. This gives

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{1}{\triangle x^2}\left[T(x+\triangle x) - 2T(x) + T(x-\triangle x)\right]. \tag{III.5}$$

Let T_k^n represent the temperature at time $n\triangle t$ and spatial location $k\triangle x$. Using this notation, $T_k^n \equiv T(k\triangle x, n\triangle t)$. Then, in two dimensions, equation (III.5) is written as

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{k+1}^n - 2T_k^n + T_{k-1}^n}{\triangle x^2}.$$

Other derivatives can be approximated in similar fashion. For example, subtracting equation (III.4) from (III.3) and neglecting the terms of order $\triangle x^3$ and higher gives

$$\frac{\partial T}{\partial x} \approx \frac{1}{2\triangle x}\left[T(x+\triangle x) - T(x-\triangle x)\right]. \tag{III.6}$$

Equation (III.6) is called a *central difference* approximation.

For the time derivative (holding the spatial term constant), $\partial T/\partial t$ may be approximated in several ways. The *forward difference* is given as

$$\frac{\partial T}{\partial t} = \frac{T_k^{n+1} - T_k^n}{\triangle t}.$$

The *backward difference* is

$$\frac{\partial T}{\partial t} = \frac{T_k^n - T_k^{n-1}}{\Delta t}.$$

The *central difference* is derived in a manner similar to that leading to equation (III.6). This results in

$$\frac{\partial T}{\partial t} = \frac{T_k^{n+1} - T_k^{n-1}}{2\Delta t}.$$

Using the forward finite difference approximation, the one-dimensional diffusion equation (III.2) can be approximated by

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = \frac{T_{k+1}^n - 2T_k^n + T_{k-1}^n}{\Delta x^2}. \tag{III.7}$$

Letting $r = \Delta t / \Delta x^2$, this equation can be written as

$$T_k^{n+1} = rT_{k-1}^n - (1 - 2r)T_k^n + rT_{k+1}^n. \tag{III.8}$$

The unknown T_k^{n+1} can now be computed explicitly at the $(n+1)$st time step using the known values of the nth time step.

While this explicit relation provides a simple means to compute unknown values, the process is only stable for $0 < \Delta t/(\Delta x)^2 \leq 1/2$ [Ref. 2]. By imposing a stability limit on $\Delta t/(\Delta x)^2$, explicit finite difference methods for the diffusion equation can give useful approximations. However, the requirement that $\Delta t/(\Delta x)^2 < 1/2$ places a severe restriction on the interval size in the t direction resulting in an increase in computation. A better method is needed.

An implicit formula is one in which two or more unknown values in the $(n+1)$st time level are specified in terms of known values in the nth time level by a single equation.

If there are M unknown values in the $(n+1)$st time level, the formula must be applied M times across the length of the time level. The resulting system of M simultaneous equations specifies the M values implicitly. A method from Crank and Nicolson [Ref. 2] combines a central difference approximation to the time derivative at the half time-step $(n+\frac{1}{2})$ with the average of central difference approximations to $\frac{\partial^2 T}{\partial x^2}$ at times $n\Delta t$ and $(n+1)\Delta t$ to approximate the equation

$$\left(\frac{\partial T}{\partial t}\right)_k^{n+\frac{1}{2}} = \left(\frac{\partial^2 T}{\partial x^2}\right)_k^{n+\frac{1}{2}} . \tag{III.9}$$

This equation is then approximated by

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = \frac{1}{2}\left(\frac{T_{k+1}^{n+1} - 2T_k^{n+1} + T_{k-1}^{n+1}}{\Delta x^2} + \frac{T_{k+1}^n - 2T_k^n + T_{k-1}^n}{\Delta x^2}\right). \tag{III.10}$$

This reduces to

$$-rT_{k+1}^{n+1} + (2+2r)T_k^{n+1} - rT_{k-1}^{n+1} = rT_{k-1}^n + (2-2r)T_k^n + rT_{k+1}^n \tag{III.11}$$

where, again, $r = \Delta t/(\Delta x)^2$. There are now three unknowns at the $(n+1)$st time step written in terms of three known values at the nth time step.

Although Crank and Nicolson's method is stable for all positive values of r, large values of r, e.g., 40, have been shown to introduce oscillations in the solution [Ref. 3]. To combat this problem, a more general finite difference approximation is implemented, using a weighted average of the finite difference approximations to $\frac{\partial^2 T}{\partial x^2}$ at the nth and $(n+1)$st time steps. This weighted average is given by

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = \alpha\left(\frac{T_{k+1}^{n+1} - 2T_k^{n+1} + T_{k-1}^{n+1}}{\Delta x^2}\right) + (1-\alpha)\left(\frac{T_{k+1}^n - 2T_k^n + T_{k-1}^n}{\Delta x^2}\right), \tag{III.12}$$

where $0 \le \alpha \le 1$. Notice that $\alpha = 0$ gives the explicit scheme, $\alpha = 1$ gives a fully implicit backward difference method, and $\alpha = 1/2$ gives the Crank-Nicolson method. This weighted

average is stable and convergent for all r if $\frac{1}{2} \leq \alpha \leq 1$, but is stable and convergent for $0 \leq \alpha \leq \frac{1}{2}$ only when $r \leq 1/(2-4\alpha)$ [Ref. 3].

Following the method outlined above, a general finite difference approximation for the time derivative of the two-dimensional conduction problem (III.1)

$$\frac{\partial T}{\partial t} = \kappa \nabla^2 T \tag{III.13}$$

is given by

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha \kappa \nabla^2 T^{n+1} + (1-\alpha)\kappa \nabla^2 T^n. \tag{III.14}$$

In this approximation, T^n is the temperature at the nth time step. The current time step is designated by n and the next time step by $(n+1)$. Combining terms for the current and subsequent time steps yields the semi-discrete relationship,

$$\left(\frac{1}{\Delta t} - \alpha\kappa\nabla^2\right) T^{n+1} = \left(\frac{1}{\Delta t} + (1-\alpha)\kappa\nabla^2\right) T^n. \tag{III.15}$$

The above conditions on α also apply here.

B. FINITE VOLUME ELEMENT METHODS

Having dealt with the time discretization, it is now necessary to address the discretization of the spatial derivatives. This development of discretization for spatial derivatives closely follows McCormick [Ref. 4]. The spatial portion of the problem is discretized using the Finite Volume Element method (FVE). The classical finite volume method (FV) is commonly used as a discretization method for computational fluid dynamics applications because of its ability to be faithful to the physics in general and conservation in particular. However, use of FV requires a scheme for approximating certain fluxes, which is often done in

an effective but rather ad hoc and restrictive way. Thus, the Finite Volume Element method was developed in an attempt to use the flexibility of the finite element representation of the unknown functions to create a more systematic FV methodology [Ref. 4]. The central idea is to approximate the discrete fluxes needed in FV by replacing the unknown PDE solution by a finite element approximation.

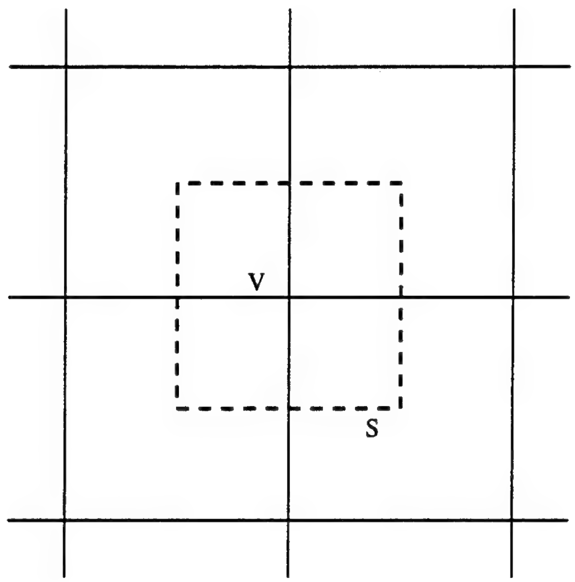

Figure 2. *Control volume V (dashed lines) and surface S.*

As an example, consider the potential flow problem in Euclidean two-dimensional space:

$$\begin{aligned} \nabla \cdot (\rho \nabla \psi) &= \eta \quad \text{in } \Omega \\ \psi &= \psi_0 \quad \text{on } \partial\Omega_N \bigcup \partial\Omega_E \text{ (Dirichlet)} \\ (\rho \nabla \psi) \cdot \hat{n} &= \psi_1 \quad \text{on } \partial\Omega_W \bigcup \partial\Omega_S \text{ (Neumann).} \end{aligned} \tag{III.16}$$

The domain for this problem is the unit square $\Omega = [0,1] \times [0,1] \in \Re^2$. In this domain, $\partial\Omega_N$ and $\partial\Omega_E$ are the northern and eastern Dirichlet boundaries of the domain Ω, and $\partial\Omega_S$

and $\partial\Omega_W$ are the southern and western Neumann boundaries. Beginning with a "control volume" V in the domain Ω with surface S (Figure 2), the potential equation (III.16) is integrated over the volume V to get

$$\int_V \nabla \cdot (\rho \nabla \psi) dV = \int_V \eta dV. \tag{III.17}$$

Using the Gauss Divergence Theorem, the left-hand side volume integral is transformed into

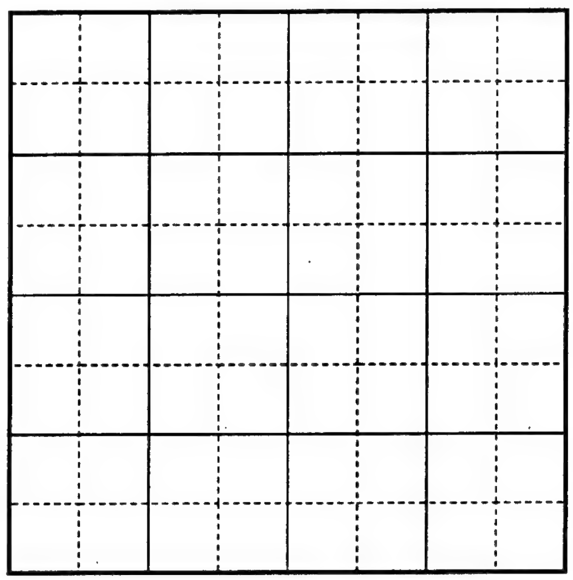

Figure 3. *Volume partition (dashed lines) of Ω.*

a surface integral,

$$\int_S (\rho \nabla \psi) \cdot \hat{n} dS = \int_V \eta dV. \tag{III.18}$$

Because this problem is two-dimensional, "volume" refers to "area" and "surface" refers to "boundary". In other words, volume integrals in two dimensions are area integrals, and the surface integrals are line integrals taken over the bounding curves.

Each side of equation (III.18) represents a flow rate in mass per unit time, and $(\rho \nabla \psi) \cdot \hat{n}$ represents the flux density across the surface S. The integral condition (III.18)

can be interpreted as a conservation law for the volume V. Conservation can be enforced over the domain by requiring (III.18) to hold over the individual volumes. The potential equation (III.16) can then be discretized by choosing a finite set of volumes that partitions Ω into volumes (Figure 3) with the integral condition in (III.18) being imposed on each volume.

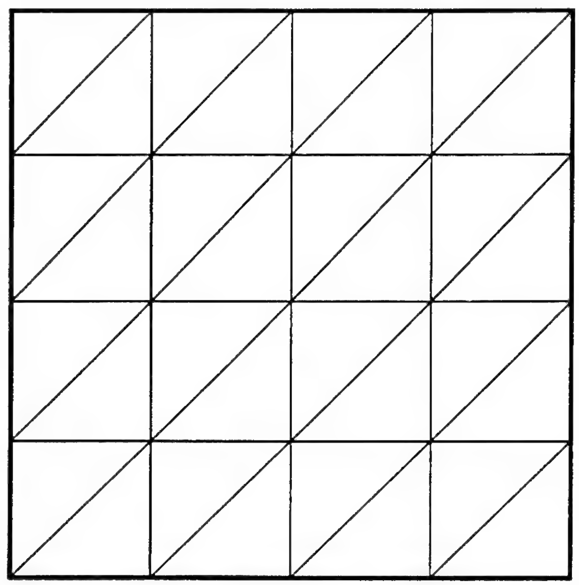

Figure 4. *Finite element partition of the domain Ω.*

It is now necessary to select the functions to be used to approximate ψ. It is well known that piecewise linear functions are often effective at approximating functions in finite element methods. Thus, Ψ can be approximated by continuous, piecewise linear functions. Then, a basis — a set of functions whose linear combinations determine the desired space of functions — is selected. The directions of interest are x and y with a uniform grid of step size $h = \frac{1}{N}$ applied in both directions. Each grid square is divided into two triangular elements by a diagonal oriented in the direction of increasing x and y (Figure 4), and "hat" functions are constructed using these elements. The FVE method replaces the exact solution ψ^* with

a finite element approximation ψ expressed as linear combinations of the basis functions. At each grid point, the value of the approximation equals the weight for the associated hat function. Thus, ψ is expressed in terms of a basis:

$$\psi = \sum_{i,j=1}^{N} u_{i,j}\phi_{i,j}(x,y), \tag{III.19}$$

where double subscripts, i, j, varying from 0 to N, are used to describe the location of the grid point (x_i, y_j). Then, $u_{i,j}$ is the value of ψ at the (i,j)th grid point, (x_i, y_j), and $\phi_{i,j}$ is the "hat" function, a continuous piecewise linear function associated with the (i,j)th grid point.

To simplify notation, the sampled values of the unknown ψ, currently written as an $(N+1) \times (N+1)$ two-dimensional array, can be considered a one-dimensional array. To accomplish this, the elements $\psi_{i,j}$ are labeled in lexicographic order. For example, the elements in the two-dimensional case are written as

$$\begin{pmatrix} \psi_{N,0} & \psi_{N,1} & \cdots & \psi_{N,N} \\ \vdots & \vdots & \vdots & \vdots \\ \psi_{1,0} & \psi_{1,1} & \cdots & \psi_{1,N} \\ \psi_{0,0} & \psi_{0,1} & \cdots & \psi_{0,N} \end{pmatrix}.$$

These elements can be written as

$$(\psi_{0,0}, \psi_{0,1}, \ldots, \psi_{0,N}, \psi_{1,0}, \psi_{1,1}, \ldots, \psi_{1,N}, \ldots, \psi_{N,0}, \psi_{N,1}, \ldots, \psi_{N,N})^T.$$

This array is normally considered to be a column vector. Using this notation, ψ is expressed in terms of a relabeled basis:

$$\psi = \sum_{l=1}^{(N+1)^2} u_l\phi_l(x,y). \tag{III.20}$$

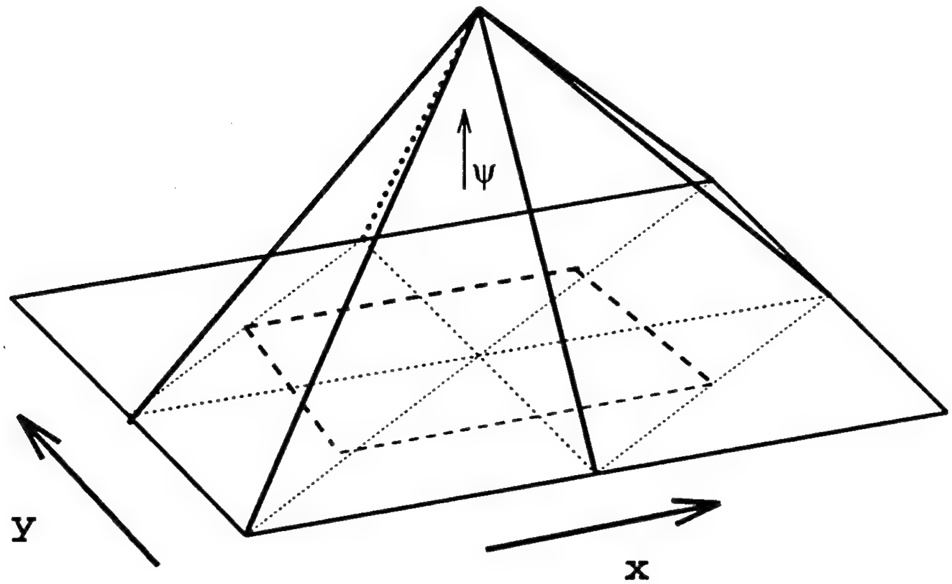

Figure 5. *Hat function associated with the (i,j)th grid point.*

Substituting (III.20) into (III.18) and integrating over each volume V_k (following the same ordering scheme) for $k = 1, 2, ...(N+1)^2$, the left side of equation (III.18) becomes, for each k,

$$\int_{S_k} \rho \nabla \psi \cdot \hat{n} dS_k = \sum_{l=1}^{(N+1)^2} u_l \int_{S_k} \rho_k \nabla \phi_l \cdot \hat{n}_k dS_k. \tag{III.21}$$

This gives the system $L\tilde{\psi} = \tilde{f}$, where

$$f_k = \int_{V_k} \eta dV_k \tag{III.22}$$

(except on the boundary) and L is the $(N+1) \times (N+1)$ matrix with entries

$$L_{k,l} = \int_{S_k} (\rho_k \nabla \phi_l) \cdot \hat{n}_k dS_k. \tag{III.23}$$

To correctly implement the boundary conditions, the system must be modified. Since both Neumann and Dirichlet conditions will be of interest, both cases will be examined.

Neumann conditions are imposed indirectly on ψ by substituting the flux value ψ_1 from the boundary conditions into the appropriate term in equation (III.18). The Neumann conditions in equation (III.16) give

$$\int_S \rho\nabla\psi \cdot \hat{n} dS = \int_{S_W \bigcup S_S} \psi_1 dS + \int_{S_N \bigcup S_E} (\rho\nabla\psi) \cdot \hat{n} dS = \int_V \eta dV, \tag{III.24}$$

where S_W and S_S represent the southern and western bounding curves of the volume where Neumann conditions are imposed.

By contrast, Dirichlet conditions in equation (III.16) are imposed directly on ψ. Thus, ψ takes on the value of ψ_0 at each Dirichlet grid point, or rather each grid point on $\partial\Omega_N \bigcup \partial\Omega_E$. There are potentially fewer ψ unknowns than equations since there are now volumes that do not have unknowns. This is easily avoided by expanding the volumes at the points neighboring the Dirichlet boundary (Figure 6).

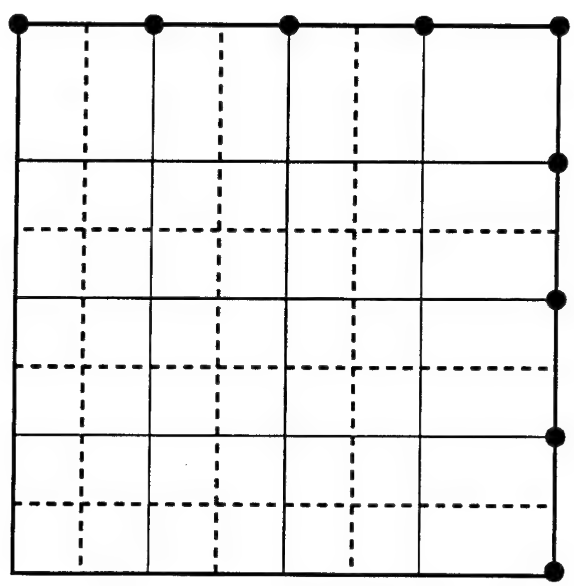

Figure 6. *Modified volume partition of Ω to accomodate Dirichlet points ($\bullet$).*

In addition, the intersection of $\partial\Omega_N$ and $\partial\Omega_W$ and the intersection of $\partial\Omega_S$ and $\partial\Omega_E$

are treated as Dirichlet grid points. These points could also have been treated as Neumann grid points. For full Neumann problems, the original partition of Figure 3 is used.

It is now time to evaluate the integrals in (III.22) and (III.23). For the integral in (III.23), the rule

$$\int_T (\rho \nabla \phi) \cdot \hat{n} dS \approx \rho(P_T) \int_T \nabla \phi \cdot \hat{n} dS \tag{III.25}$$

is used on each bounding segment, T, of the surface. That is, T is one of S_N, S_E, S_S or S_W, and P_T is the point of intersection of T and the grid lines passing through N_V, the grid point at the center of the volume. For Neumann boundary segments S_S and S_W, the rule is

$$\int_T \psi_1 dS \approx \psi_1(M_T)|T|, \tag{III.26}$$

where M_T is the midpoint of T and $|T|$ is the "surface area," or length, of T.

Consider the discretization on a uniform mesh with grid size $h = \frac{1}{N}$ in both coordinates. Returning to the double subscript notation, i and j vary from 0 to $N-1$, where 0 corresponds to the Neumann boundary grid point, or node, along $\partial\Omega_W$ and $\partial\Omega_S$, and $N-1$ corresponds to the nodes next to the Dirichlet boundary, $\partial\Omega_N$ and $\partial\Omega_E$. For a general interior node ($i > 0, j < m-1$), consider one segment of the boundary, say the NW side (Figure 7). To evaluate the integral

$$\rho(P_T) \int_{NW} \nabla \psi \cdot \hat{n} dS, \tag{III.27}$$

first notice that, with $\hat{n} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$,

$$\nabla \psi \cdot \hat{n} = \psi_y = \frac{\psi_{i,j+1} - \psi_{i,j}}{h}, \tag{III.28}$$

since ψ is piecewise linear. Then,

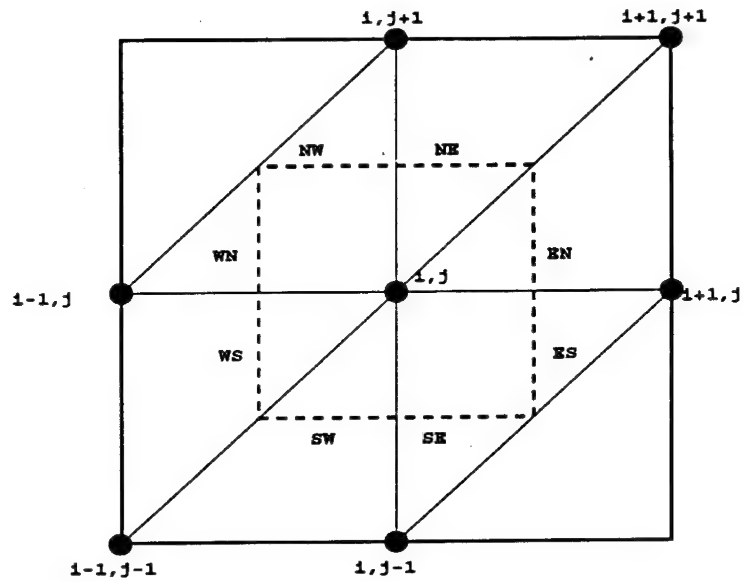

Figure 7. *Volume of general interior grid point* (i, j).

$$\int_{NW} \nabla\psi \cdot \hat{n} dS = \frac{\psi_{i,j+1} - \psi_{i,j}}{h} \int_{NW} dS = \frac{h}{2}\left(\frac{\psi_{i,j+1} - \psi_{i,j}}{h}\right) = \frac{\psi_{i,j+1} - \psi_{i,j}}{2}. \quad \text{(III.29)}$$

This process is repeated on all segments of the volume boundary (8 calculations in all). The NW and NE calculations can be combined into a single North calculation. The same occurs for the South, East and West sides:

$$\int_{N} \nabla\psi \cdot \hat{n} dS = \psi_{i,j+1} - \psi_{i,j} \quad \text{(III.30)}$$

$$\int_{S} \nabla\psi \cdot \hat{n} dS = \psi_{i,j-1} - \psi_{i,j} \quad \text{(III.31)}$$

$$\int_{E} \nabla\psi \cdot \hat{n} dS = \psi_{i+1,j} - \psi_{i,j} \quad \text{(III.32)}$$

$$\int_{W} \nabla\psi \cdot \hat{n} dS = \psi_{i-1,j} - \psi_{i,j} \quad \text{(III.33)}$$

Then, for the general interior grid point, or node,

$$\rho(P_T) \int_{S_j} \nabla\psi \cdot \hat{n} dS = \rho(P_W)\psi_{i-1,j} + \rho(P_E)\psi_{i+1,j} + \rho(P_N)\psi_{i,j+1} + \rho(P_S)\psi_{i,j-1} - \sum \psi_{i,j}, \quad \text{(III.34)}$$

where $\sum = \rho(P_N) + \rho(P_S) + \rho(P_E) + \rho(P_W)$.

The quadrature rule is used on the integral in (III.22). This results in

$$\int_{V_j} \eta dV \approx \eta_{ij} |V_j|, \tag{III.35}$$

where η_{ij} is the value of $\eta(x, y)$ associated with V_j and $|V_j| = \int_{V_j} dV$ is the "volume," or area, of V_j. For a general interior node, $|V_j| = h^2$. Thus, for a general interior node, the stencil equation is

$$\begin{bmatrix} & \rho_{i,j+\frac{1}{2}} & \\ \rho_{i-\frac{1}{2},j} & -\sum & \rho_{i+\frac{1}{2},j} \\ & \rho_{i,j-\frac{1}{2}} & \end{bmatrix} \psi_{ij} = h^2 \eta(ih, jh), \tag{III.36}$$

where Σ signifies the sum of the outer stencil entries and $\rho_{i,j} = \rho(ih, jh)$. This notation for $\rho(ih, jh)$ will also be used in subsequent stencils. In the stencil notation,

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \psi_{i,j},$$

stencil entries (A, B, ...) indicate the value to be applied, with the position of the entry indicating to which grid point the value is to be applied. Specifically, the center of the stencil is the current grid point, with the surrounding stencil entries corresponding to the neighboring grid points in those directions on the grid. Thus, the above stencil has the value

$$A\psi_{i-1,j+1} + B\psi_{i,j+1} + C\psi_{i+1,j+1} + D\psi_{i-1,j} + E\psi_{i,j} + F\psi_{i+1,j} + \ldots.$$

For a side Neumann node, stencil development follows a similar path. Along the boundary, recall that the Neumann condition gives $\nabla\psi \cdot \hat{n} = \psi_1$. Integrating on the Southern boundary (Figure 8) and using the midpoint rule,

$$\int_{\partial\Omega} \psi_1(x, y) dS = h\psi_1(x_i, 0). \tag{III.37}$$

For the smaller volume,

$$\int_V \eta dV = \eta_{ij}|V_j| = \eta_{ij}\left(\frac{h^2}{2}\right). \tag{III.38}$$

For a side Neumann boundary grid point ($j = 0$ on southern boundary), the stencil is

$$\begin{bmatrix} & \rho_{i,\frac{1}{2}} & \\ \frac{1}{2}\rho_{i-\frac{h}{2},0} & -\sum & \frac{1}{2}\rho_{i+\frac{1}{2},0} \\ & & \end{bmatrix} \psi_{i0} = \frac{h^2}{2}\eta(ih,0) - h\psi_1(ih,0). \tag{III.39}$$

For the corner Neumann node ($i = j = 0$), the stencil is developed in the same manner as

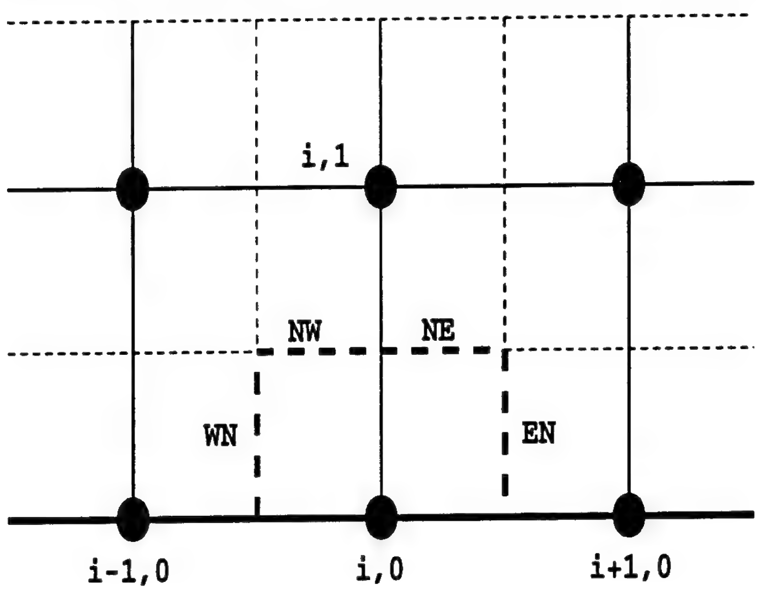

Figure 8. *Side Neumann grid point on the southern boundary* ($j = 0$).

the side Neumann nodes. The stencil is

$$\begin{bmatrix} & \frac{1}{2}\rho_{0,\frac{1}{2}} & \\ & -\sum & \frac{1}{2}\rho_{\frac{h}{2},0} \\ & & \end{bmatrix} \psi_{00} = \frac{h^2}{4}\eta(0,0) - \frac{h}{2}\left(\psi_1\left(0,\frac{h}{4}\right) + \psi_1\left(\frac{h}{4},0\right)\right). \tag{III.40}$$

The stencils for the neighboring Dirichlet grid points ($i = N - 1$ or $j = N - 1$) are the same as at interior grid points, except that the Dirichlet points are eliminated and the stencil entry reaching to a Dirichlet node value is removed and placed on the right-hand side as a coefficient of the boundary data. There are two possible stencils for a neighboring

Dirichlet node: the reduced volume and the extended volume. For the neighboring Dirichlet nodes of the Eastern boundary, $\partial\Omega_E$ $(i = N-1, 0 < j < N-1)$, the reduced volume stencil is

$$\begin{bmatrix} & \rho_{N-1,j+\frac{1}{2}} & \\ \rho_{N-\frac{3}{2},j} & -\sum & \\ & \rho_{N-1,j-\frac{1}{2}} & \end{bmatrix} \psi_{N-1,j} = h^2\eta(1-h, jh) - \rho_{N-\frac{h}{2},j}\psi_0(1, jh), \tag{III.41}$$

where Σ is the sum of the outer stencil entries before the boundary terms are removed. For this case, $\Sigma = \rho(1-\frac{h}{2}, jh) + \rho(1-\frac{3h}{2}, jh) + \rho(1-h, (j-\frac{1}{2})h) + \rho(1-h, (j+\frac{1}{2})h)$.

The stencil for the same node in an extended volume is

$$\begin{aligned} \begin{bmatrix} & \frac{1}{2}\rho_{N-1,j+\frac{1}{2}} & \\ \rho_{N-\frac{3}{2},j} & -\sum & \\ & \rho_{N-1,j-\frac{1}{2}} & \end{bmatrix} \psi_{N-1,j} &= \frac{3h^2}{2}\eta(1-h, jh) + \frac{1}{2}\rho_{N-\frac{1}{2},j}\psi_0(1, jh) \\ &\quad -\rho_{N,j}\psi_0(1, (j+1)h) - \frac{1}{2}\rho_{N,j}\psi_0(1, (j-1)h) \end{aligned}$$

where Σ is, again, the sum of the entries of the stencil without the boundary terms removed. The extended volume stencil for the neighboring corner Dirichlet point ($i = N-1$ and $j = N-1$) is

$$\begin{aligned} \begin{bmatrix} \frac{1}{2}\rho_{N-\frac{3}{2},N-1} & -\sum & \\ & \frac{1}{2}\rho_{N-1,N-\frac{3}{2}} & \end{bmatrix} \psi_{N-1,N-1} &= \frac{9h^2}{4}\eta(1-h, 1-h) - \frac{1}{2}\rho_{N-\frac{1}{2},N-1}\psi_0(1, 1-h) \\ &\quad -\rho_{N-\frac{1}{2},N-\frac{3}{2}}\psi_0(1, 1-2h) - \frac{1}{2}\rho_{N-1,N-\frac{1}{2}}\psi_0(1-h, 1) \\ &\quad -\rho_{N-\frac{3}{2},N-\frac{1}{2}}\psi_0(1-2h, 1). \end{aligned}$$

C. DISCRETIZATION OF THE WELDPOOL PROBLEM

The technique used on the potential problem (III.16) may be used to discretize the weldpool problem. To begin, the FVE stencils for the conduction problem (II.5) are devel-

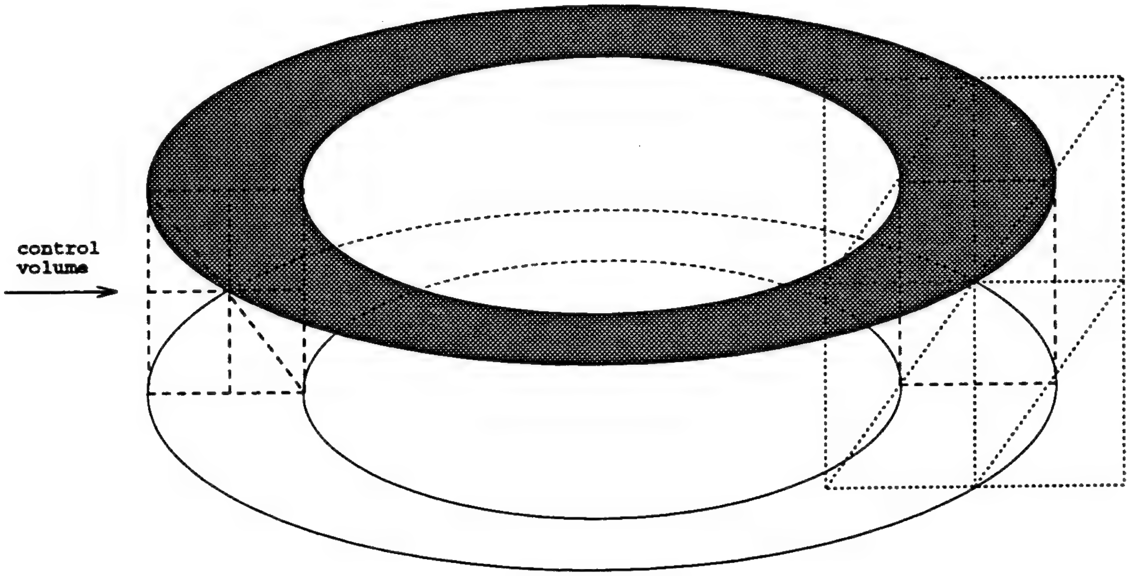

Figure 9. *Toroidal volume for the conduction problem.*

oped. For the axisymmetric conduction equation, a control volume for the lth grid point is a toroidal prism. This volume, V_l, is generated by taking a two-dimensional volume for the lth grid point in the (r, z) plane and sweeping the volume through 360° in the φ direction (Figure 9). Previously, a general finite difference semi-discrete approximation for the equation

$$\frac{\partial T}{\partial t} = Ma^{-1}\nabla^2 T$$

was found to be

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha Ma^{-1}\nabla^2 T^{n+1} + (1-\alpha)Ma^{-1}\nabla^2 T^n.$$

Combining terms for the current and subsequent time steps yielded the semi-discrete relationship,

$$\left(\frac{1}{\Delta t} - \alpha Ma^{-1}\nabla^2\right) T^{n+1} = \left(\frac{1}{\Delta t} + (1-\alpha)Ma^{-1}\nabla^2\right) T^n. \qquad \text{(III.42)}$$

To complete the discretization, discrete equations for the spatial derivatives are developed. The resulting equations are assembled into an algebraic system. Begin by integrating (III.42)

over all of the control volumes to yield

$$\int_V \left(\frac{1}{\Delta t} - \alpha Ma^{-1}\nabla^2\right) T^{n+1} dV = \int_V \left(\frac{1}{\Delta t} + (1-\alpha) Ma^{-1}\nabla^2\right) T^n dV. \tag{III.43}$$

Here, V represents the partition of the domain Ω into the smaller control volumes, V_l, or $V = \bigcup V_l$. Applying the Gauss Divergence Theorem gives

$$\frac{1}{\Delta t}\int_V T^{n+1} dV - \alpha Ma^{-1}\int_S \nabla T^{n+1}\cdot\hat{n} dS = \frac{1}{\Delta t}\int_V T^n dV + (1-\alpha) Ma^{-1}\int_S \nabla T^n\cdot\hat{n} dS, \tag{III.44}$$

where $\hat{n}$ is the outward normal.

As in the model problem, T can be approximated by continuous, piecewise linear functions. Using cylindrical coordinates, a basis is chosen for the space in which these functions are found. Because of cylindrical symmetry, the discretization can be examined on the half plane $\theta = 0$. Hence, T is treated as though it is a function defined on a two-dimensional domain in the (r, z) plane. In this two-dimensional domain, "hat" functions , $\phi_{i,j}(r, z)$, are chosen as the basis functions, and the unknown function T is approximated as

$$T(r, z) = \sum_{i,j=0}^{N} T_{i,j}\phi_{i,j}(r, z), \tag{III.45}$$

where the uniform grid step size is $h = 1/N$.

A change of notation is again used. The sampled values of the unknown T, now written as an $(N+1)\times(N+1)$ two-dimensional array, can be considered a one-dimensional array. To accomplish this, the elements $T_{i,j}$ are labeled in lexicographic order from $l = 1$ to $l = (N+1)^2$. For example, the elements in the two-dimensional case are written as

$$\begin{pmatrix} T_{N,0} & T_{N,1} & \cdots & T_{N,N} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1,0} & T_{1,1} & \cdots & T_{1,N} \\ T_{0,0} & T_{0,1} & \cdots & T_{0,N} \end{pmatrix}.$$

These elements can be written lexicographically in a column vector as

$$(T_{0,0}, T_{0,1}, \ldots, T_{0,N}, T_{1,0}, T_{1,1}, \ldots, T_{1,N}, \ldots, T_{N,0}, T_{N,1}, \ldots, T_{N,N})^T .$$

The values on the grid Ω are given now as

$$T_l = T(r_l, z_l).$$

Substituting equation (III.45) into equation (III.44) gives

$$\begin{aligned}\sum_{l=1}^{(N+1)^2} T_l^{n+1}\left[\frac{1}{\Delta t}\int_V \phi_l^{n+1} dV \right. & - \left. \alpha Ma^{-1}\int_S \nabla\phi_l^{n+1}\cdot\hat{n}dS\right] \\ & = \sum_{l=1}^{(N+1)^2} T_l^n\left[\frac{1}{\Delta t}\int_V \phi_l^n dV + (1-\alpha)Ma^{-1}\int_S \nabla\phi_l^n\cdot\hat{n}dS\right].\end{aligned} \tag{III.46}$$

The volume V of the domain is partitioned into $l = (N+1)^2$ volumes (actually, there are fewer volumes because of boundary conditions) where

$$V = \bigcup V_l \quad \text{and} \quad V_l \bigcap V_k = 0 \quad \text{for} \quad l \neq k.$$

Substituting the partitioned volumes into equation (III.46) results in

$$\begin{aligned}\sum_{l=1}^{(N+1)^2} T_l^{n+1}\left[\frac{1}{\Delta t}\sum_{k=1}^{(N+1)^2}\int_{V_k}\phi_l^{n+1}dV_k - \alpha Ma^{-1}\sum_{k=1}^{(N+1)^2}\int_{S_k}\nabla\phi_l^{n+1}\cdot\hat{n}dS_k\right] \\ = \sum_{l=1}^{(N+1)^2} T_l^n\left[\frac{1}{\Delta t}\sum_{k=1}^{(N+1)^2}\int_{V_k}\phi_l^n dV_k + (1-\alpha)Ma^{-1}\sum_{k=1}^{(N+1)^2}\int_{S_k}\nabla\phi_l^n\cdot\hat{n}dS_k\right].\end{aligned} \tag{III.47}$$

Notice here that the interior surface integrals cancel out. The eastern edge of one interior volume coincides with the western edge of the neighboring volume. This edge is integrated over twice, but in opposing directions, resulting in the cancellation.

This set of equations can be written as an operator equation, $L[T^{n+1}] = f(T^n)$, or as a matrix equation, $\mathbf{M}_1[\vec{T}^{n+1}] = \mathbf{M}_2[\vec{T}^n]$. The operator L and the elements of the matrix $\mathbf{M}_1$

are given as

$$L = \int_V \left(\frac{1}{\Delta t} - \alpha Ma^{-1} \nabla^2 \right) dV, \tag{III.48}$$

$$\mathbf{M}_{k,l} = \frac{1}{\Delta t} \int_{V_k} \phi_l^{n+1} dV_k - \alpha Ma^{-1} \int_{S_k} \nabla \phi_l^{n+1} \cdot \hat{n} dS_k, \tag{III.49}$$

while $f(T^n)$ and the elements of $\mathbf{M}_2$ are given as

$$f(T^n) = \int_V \left(\frac{1}{\Delta t} + (1-\alpha) Ma^{-1} \nabla^2 \right) T^n dV \tag{III.50}$$

$$\mathbf{M}_{k,l} = \frac{1}{\Delta t} \int_{V_k} \phi_l^n dV_k + (1-\alpha) Ma^{-1} \int_{S_k} \nabla \phi_l^n \cdot \hat{n} dS_k. \tag{III.51}$$

Any function whose coefficients satisfy the resulting set of discrete equations also satisfies the conservation law over any volume made up of the union of control volumes. However, complications arise when considering the boundaries. The Neumann conditions on the western and southern boundaries are incorporated indirectly into T by substituting the normal derivative value into the appropriate term in equation (III.44). The Dirichlet conditions on the northern and eastern boundaries are imposed directly on T. The control volumes are extended so that the control volumes normally associated with the boundary points are taken into account by the extended volume of the neighboring points (Figure 10). The integrals over the basis functions are calculated by representing the basis functions as collections of triangular planes. These triangular planes are assembled to form the hat function (Figure 5). An interior grid point has six triangular planes joined in this manner, each triangular plane corresponding to the six triangles surrounding the grid point (Figure 7). Integration results in the following stencils for a general interior point on a uniform grid:

$$\int_{V_{i,j}} \left(\sum_{p,q} T_{p,q} \phi_{p,q} \right) dV = \frac{h^2 r}{384} \begin{bmatrix} & 32-5\epsilon & 16+5\epsilon \\ 32-11\epsilon & 224 & 32+11\epsilon \\ 16-5\epsilon & 32+5\epsilon & \end{bmatrix} T_{i,j}$$

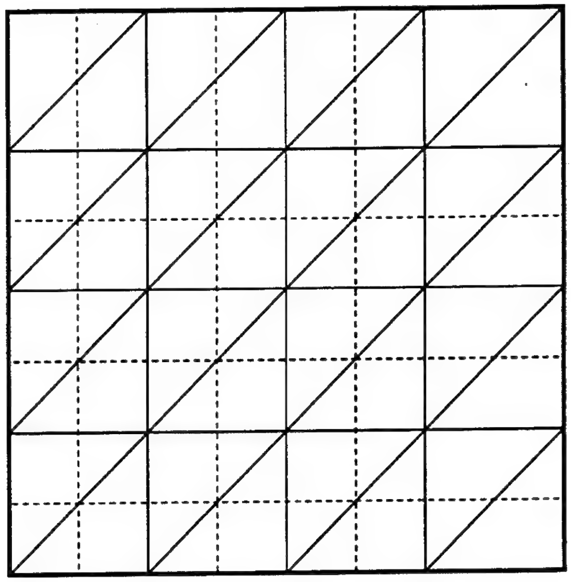

Figure 10. *A cross section in the (r,z) plane of a finite volume element grid with the domain partitioned into control volumes (dashed lines). Extended control volumes are used for grid points neighboring Dirichlet points. The triangular finite elements (solid diagonal lines) form the piecewise linear "hat" function centered at each node.*

$$\int_{S_{i,j}} \left(\sum_{p,q} T_{p,q} \nabla \phi_{p,q} \cdot \hat{n} \right) dS \;=\; \frac{r}{2} \begin{bmatrix} & 2 & \\ 2-\epsilon & -8 & 2+\epsilon \\ & 2 & \end{bmatrix} . T_{i,j},$$

where $\epsilon \equiv \frac{h}{r}$ (r is the radial coordinate for the central point of the volume). The stencils use the two-dimensional labeling. The summations signify summing over all "hat" functions for each volume. Also, the volume and surface integrals are in cylindrical coordinates. This means that the integrals are as follows:

$$\int_V dV = \int_V r dr d\theta dz = 2\pi \int_A r dr dz$$

$$\int_S dS = \int_S r d\theta dl = 2\pi \oint r dl.$$

In each case, the 2π resulting from the integration in the θ direction has been factored out.

The discretization of the convection-diffusion equation,

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = Ma^{-1} \nabla^2 T,$$

has the same form as the conduction equation for the solid except for the term

$$\int_V \nabla \cdot (\mathbf{u}T)dV.$$

Using the Gauss Divergence Theorem, this portion of the equation is also discretized. By cylindrical symmetry,

$$\int_V \nabla \cdot (\mathbf{u}T)dV = \int_{\partial V} (\mathbf{u}T) \cdot \hat{n}dA = 2\pi \oint (\mathbf{u}T) \cdot \hat{n}rdl,$$

where the 2π, as above, is factored out. Remembering that

$$\mathbf{u} = -\frac{\mathbf{1}}{\mathbf{r}}\frac{\partial \mathbf{\Psi}}{\partial \mathbf{z}}\hat{\mathbf{r}} + \frac{\mathbf{1}}{\mathbf{r}}\frac{\partial \mathbf{\Psi}}{\partial \mathbf{r}}\hat{\mathbf{z}}$$

produces the result

$$\oint (\mathbf{u}T) \cdot \hat{n}rdl = \int_N \frac{\partial \Psi}{\partial r} T dr - \int_S \frac{\partial \Psi}{\partial r} T dr + \int_W \frac{\partial \Psi}{\partial z} T dz - \int_E \frac{\partial \Psi}{\partial z} T dz.$$

Substituting the piecewise linear "hat" functions for the unknowns Ψ and T into the above integral results in discrete equations. Evaluating and summing the integrals and using stencil notation gives

$$\int_V \nabla \cdot (\mathbf{u}T)dV =$$

$$\frac{1}{8}\begin{pmatrix} & \begin{pmatrix} & & 1 \\ -2 & 1 & \\ & & \end{pmatrix}\Psi & \begin{pmatrix} & -1 & \\ & & 1 \\ & & \end{pmatrix}\Psi \\ \begin{pmatrix} & 2 & \\ & -1 & \\ -1 & & \end{pmatrix}\Psi & \begin{pmatrix} & -1 & \\ 1 & & 1 \\ & -1 & \end{pmatrix}\Psi & \begin{pmatrix} & & -1 \\ & -1 & \\ & 2 & \end{pmatrix}\Psi \\ \begin{pmatrix} & & \\ 1 & & \\ & -1 & \end{pmatrix}\Psi & \begin{pmatrix} & & \\ & 1 & -2 \\ 1 & & \end{pmatrix}\Psi & \end{pmatrix} T_{i,j}.$$

Discretization of the vorticity (equation II.7) follows the same scheme. Integrating the vorticity equation over a control volume results in

$$\frac{\partial}{\partial t}\int_V \hat{\omega}\cdot\hat{\theta}dV - \int_V \nabla\times(\mathbf{u}\times\hat{\omega})dV = -Re^{-1}\int_V \nabla\times\nabla\times\hat{\omega}dV. \tag{III.52}$$

The integration over the volume equates to an integral over the cross sectional area with $\hat{\theta}$ acting as the normal vector. Applying Stokes' theorem to (III.52) results in

$$\frac{\partial}{\partial t}\int\int_A \omega dr dz - \oint_C \hat{t}\cdot(\mathbf{u}\times\hat{\omega})dl = -Re^{-1}\oint_C \hat{t}\cdot(\nabla\times\hat{\omega})dl. \tag{III.53}$$

Again, a uniform grid with step size h in both the r and z directions is applied. Each square of the grid is divided into two triangular elements by a diagonal in the direction of increasing r and z (Figure 10) and linear interpolation is used over each triangular element. The control volume cross sections are square with side length h and centered on a grid point, except for the control volumes surrounding side and corner Neumann grid points and side and corner Dirichlet boundary points (again, see Figure 10).

The area integrals are taken over six triangular regions, while the line integrals are taken over four line segments, each line segment containing halves in two different elements. Recall that

$$\mathbf{u} = -\frac{1}{r}\frac{\partial\Psi}{\partial z}\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial\Psi}{\partial r}\hat{\mathbf{z}}, \tag{III.54}$$

and note that the vorticity vector, $\hat{\omega}$, has only one component, the component in the $\hat{\theta}$ direction. Using the expression for $\mathbf{u}$ from above results in

$$\mathbf{u}\times\hat{\omega} = -\frac{\omega}{r}\frac{\partial\Psi}{\partial r}\hat{\mathbf{r}} - \frac{\omega}{r}\frac{\partial\Psi}{\partial z}\hat{\mathbf{z}}.$$

The integrals in (III.53) then reduce to

$$\frac{\partial}{\partial t}\int\int_A \omega dr dz \;+\; \int_N \frac{\partial\Psi}{\partial r}\frac{\omega}{r}dr - \int_S \frac{\partial\Psi}{\partial r}\frac{\omega}{r}dr + \int_W \frac{\partial\Psi}{\partial z}\frac{\omega}{r}dz - \int_E \frac{\partial\Psi}{\partial z}\frac{\omega}{r}dz$$

$$
= \quad Re^{-1}\left(\int_N \frac{\partial \omega}{\partial z}dr - \int_S \frac{\partial \omega}{\partial z}dr - \int_W \frac{\partial (r\omega)}{\partial r}\frac{1}{r}dz + \int_E \frac{\partial (r\omega)}{\partial r}\frac{1}{r}dz\right).
$$

After substituting the piecewise linear element representation of the unknowns into the above integrals, the discrete equations for an interior grid point on a uniform grid are given in stencil form as

$$
\frac{d}{dt}\frac{h^2 r}{24}\begin{pmatrix} & 2 & 1 \\ 2 & 14 & 2 \\ 1 & 2 & \end{pmatrix}\omega
$$

$$
+\frac{1}{8}\left(\begin{array}{ccc}
 & \begin{pmatrix} & \frac{1}{2}\epsilon^+ & 1 \\ -2 & 1-\frac{1}{2}\epsilon^+ & \\ & & \end{pmatrix}\Psi & \begin{pmatrix} & -1 & \frac{1}{2}\epsilon^- \\ & & 1^- \\ & & \end{pmatrix}\Psi \\
\begin{pmatrix} & 2*1+ & \\ \frac{1}{2}\epsilon^+ & -1-\epsilon^+ & \\ -1+ & & \end{pmatrix}\Psi & \begin{pmatrix} & -1+\frac{1}{2}\epsilon^+ & \epsilon^- \\ 1+\epsilon^+ & -\epsilon^+ + \epsilon^- & 1-\epsilon^- \\ -\epsilon^+ & -1-\frac{1}{2}\epsilon^- & \end{pmatrix}\Psi & \begin{pmatrix} & & -1^- \\ & -1+\epsilon^- & -\frac{1}{2}\epsilon^- \\ & 2*1- & \end{pmatrix}\Psi \\
\begin{pmatrix} & & \\ 1^+ & & \\ -\frac{1}{2}\epsilon^+ & -1 & \end{pmatrix}\Psi & \begin{pmatrix} & & \\ & 1+\frac{1}{2}\epsilon^- & -2 \\ 1 & -\epsilon^- & \end{pmatrix}\Psi &
\end{array}\right)\omega
$$

$$
= r(Re^{-1})\begin{pmatrix} & 1-\frac{1}{8}\epsilon^+ & \\ 1-\frac{3}{8}\epsilon^+ & -4-\frac{3}{8}(\epsilon^+ - \epsilon^-) & 1+\frac{3}{8}\epsilon^- \\ -\frac{1}{8}\epsilon^+ & 1+\frac{1}{8}\epsilon^- & \end{pmatrix}\omega,
$$

where $\epsilon \equiv h/r$, $\epsilon^+ \equiv \epsilon/(1-\epsilon/2)$, $\epsilon^- \equiv \epsilon/(1+\epsilon/2)$, $1^+ \equiv 1/(1-\epsilon/2)$, and $1^- \equiv 1/(1+\epsilon/2)$.

The last equation to be considered for discretization is the stream function (II.8),

$$
\hat{\omega} = \nabla \times \mathbf{u}, \tag{III.55}
$$

where

$$
\mathbf{u} = \nabla \times \left(\frac{\Psi}{r}\right)\hat{\theta}.
$$

Integrating over the control volumes and applying Stokes Theorem to the right side of equation (III.55) gives

$$
\int\int_A \omega drdz = \oint_C \hat{t}\cdot \mathbf{u}dl, \tag{III.56}
$$

where $\hat{t}$ is, again, the unit vector tangent to the curve C. With $\mathbf{u}$ as above (equation III.54), these integrals reduce to

$$\int\int_A \omega dr dz = -\int_N \frac{\partial \Psi}{\partial z}\frac{1}{r}dr + \int_S \frac{\partial \Psi}{\partial z}\frac{1}{r}dr + \int_W \frac{\partial \Psi}{\partial r}\frac{1}{r}dz - \int_E \frac{\partial \Psi}{\partial r}\frac{1}{r}dz. \tag{III.57}$$

Again substituting the piecewise linear element representation of the unknowns into the above integrals gives the discrete equations for an interior grid point on a uniform grid. In stencil form,

$$\frac{d}{dt}\frac{h^2 r}{24}\begin{pmatrix} & 2 & 1 \\ 2 & 14 & 2 \\ 1 & 2 & \end{pmatrix}\omega = \begin{pmatrix} & -1 & \\ -1^{+} & (2+1^{-}+1^{+}) & -1^{-} \\ & -1 & \end{pmatrix}\Psi_{i,j}, \tag{III.58}$$

where the values for 1^{-} and 1^{+} are as above.

The weldpool problem is now discretized on a uniform grid. In the next section, the discussion will center on what a composite grid is, why a composite grid is needed, and how to resolve the weldpool problem more efficiently.

IV. FAST ADAPTIVE COMPOSITE METHODS

This chapter will introduce the concepts of fast adaptive composite (FAC) grid methods. FAC methods are used to discretize and find solutions for partial differential equations (PDEs) on a composite grid. A composite grid is formed by the union of a nested sequence of uniform grids of varying size and scale. A patch, or rectangular uniform grid, can be aligned with coarser grids and can be specified by its relative location on the coarser grid, its dimensions, and its mesh size (Figure 11). The PDE is discretized and solved on this composite grid. However, the actual computations take place on the uniform subgrids. In this manner, FAC maintains the advantages of uniform grid discretizations while allowing effective adaptation to the local phenomena. FAC methods require discretization of the PDE on both the individual levels and the composite grid. The approach used here, the finite volume element (FVE) method, combines the finite volume and finite element methods [Ref. 4].

Many physical models encountered provide excellent opportunities to employ FAC. Often, physical models require higher accuracy only in limited regions of a domain. The physical model could be resolved on a global uniform grid with the smallest required mesh size, but computational power is rarely unlimited, and some method for locally adapting resolution and accuracy is needed. It can also be an advantage to introduce as little nonuniformity into the method as possible. This is accomplished using composite grids. For an example of a coarse grid, consider the two-level grid in Figure 12. The coarse grid is defined on a domain Ω_{2h}, where $2h$ indicates the grid spacing. There are several distinct

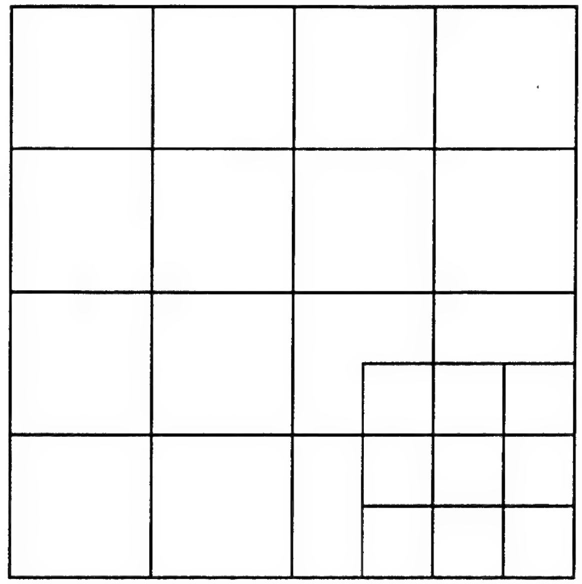

Figure 11. *A composite grid with its fine grid patch.*

types of gridpoints, such as regular grid, coarse grid interface, fine grid interface, and slave points. The fine grid points can be treated as regular interior points of the refinement patch, while coarse grid interface points and slave interface points can be taken as boundary points for the fine grid patch. Since the functions on all of the grids are piecewise linear, the slave interface points can be calculated as a linear interpolation of neighboring coarse grid interface points. Multilevel methods can then be implemented. These methods make use of fully overlapping grids of different scales and correction of the coarse grid equations by the composite grid residuals to approximate the PDE solution. This is accomplished by solving the system on the fine grid level and then correcting the coarse grid equations with the fine grid approximations.

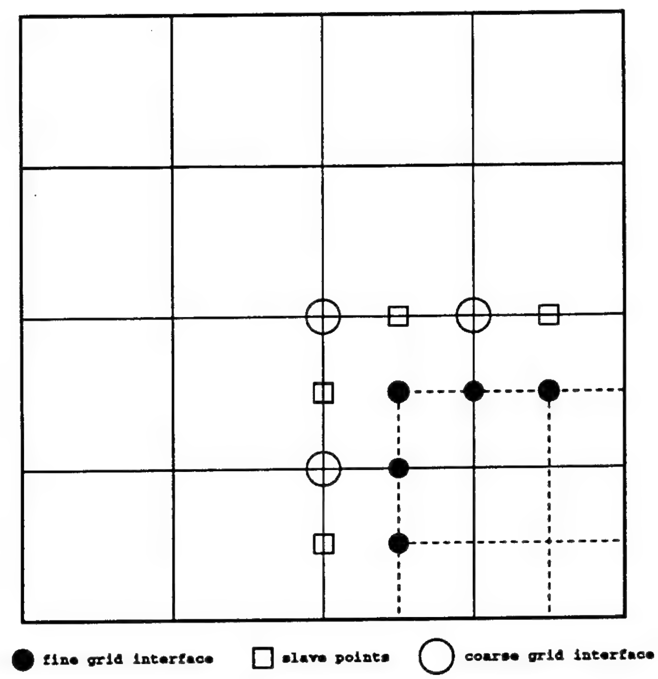

Figure 12. *Composite Grid Interface Points.*

A. FUNDAMENTALS OF MULTIGRID

Before entering into a development of FAC, the concepts of multigrid should be introduced. Multigrid is a basic component of FAC because of its apparent effectiveness as an iterative method. It is of interest to determine how multigrid behaves as a solver of discretizations on uniform grids, because multigrid will be used for composite grid equations restricted to the global and local subgrids [Ref. 4].

Multigrid methods were first developed to solve boundary value problems posed on spatial domains. These boundary value problems can be made discrete by selecting a set of grid points in a domain. The discrete problem that results is a system of algebraic equations associated with the chosen grid points. The simplicity of these boundary value problems provides a good introduction to multigrid.

1. Two Model Problems

Consider the two-point boundary value problem describing the steady-state temperature distribution in a long uniform rod. The second-order ordinary differential equation is

$$-u''(x) + \sigma u(x) = f(x), \quad 0 < x < 1, \quad \sigma \geq 0, \tag{IV.1}$$

where $u(0) = u(1) = 0$. The simplest method for solving this equation numerically is the finite difference method. The domain of the problem is partitioned into N subintervals in a step size of $h = 1/N$. A grid is established with points $x_j = jh$ (Figure 13).

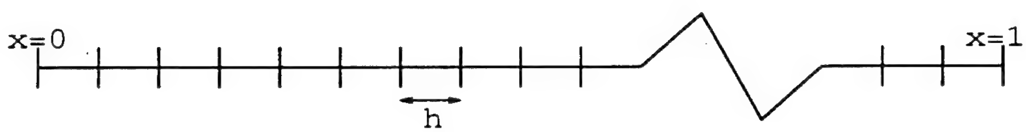

Figure 13. *One-dimensional grid on the interval* $0 \leq x \leq 1$ *with grid spacing* $h = 1/N$ *and grid point* $x_j = jh$ *for* $0 \leq j \leq N$.

At each of the $N-1$ interior grid points, the differential equation (IV.1) is replaced by a finite difference approximation. Since the exact solution is unknown, the approximation v_j is introduced. The approximate solution is represented by the vector $\mathbf{v} = (v_1, v_2, \ldots, v_{N-1})$, whose components satisfy the $N-1$ linear equations of the finite difference approximation,

$$\frac{-v_{j+1} + 2v_j - v_{j-1}}{h^2} + \sigma v_j = f(x_j), \qquad 1 \leq j \leq N-1, \tag{IV.2}$$

with the boundary conditions $v_0 = v_N = 0$. Representing this system of linear equations in matrix form gives

$$\frac{1}{h^2}\begin{bmatrix} 2+\sigma h^2 & -1 & & & \\ -1 & 2+\sigma h^2 & -1 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot \quad \cdot & -1 \\ & & & -1 & 2+\sigma h^2 \end{bmatrix}\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ v_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_{N-1} \end{bmatrix} \tag{IV.3}$$

or simply $A\mathbf{v} = \mathbf{f}$. The matrix A is tridiagonal, symmetric positive definite, sparse (the matrix elements are predominantly zero), and has dimension $(N-1) \times (N-1)$.

For the two-dimensional case, consider Poisson's equation, a second-order partial differential equation given by

$$-u_{xx} - u_{yy} = f(x,y), \quad 0 < x < 1, \quad 0 < y < 1, \tag{IV.4}$$

with $u = 0$ along the boundary of the unit square (Figure 14).

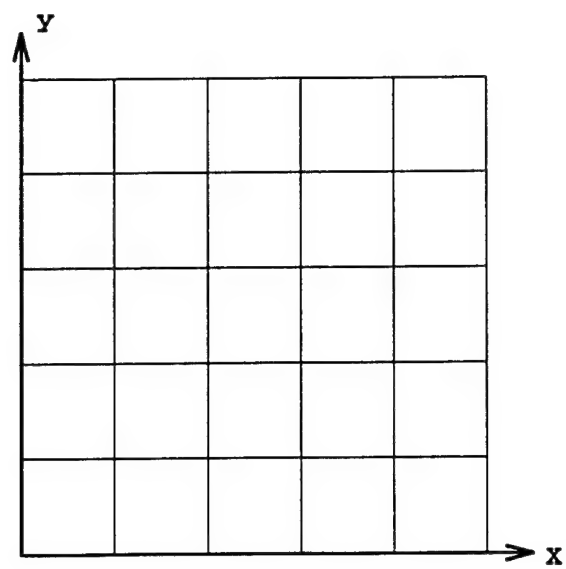

Figure 14. *Two-dimensional grid on the unit square.*

As before, Poisson's equation can be cast in a discrete form using the grid points $(x_i, y_j) = (ih, jh)$, where $h = 1/N$ is the width of the subintervals. Replacing the derivatives of Poisson's equation (IV.4) with second-order finite differences and the exact solution with the approximation $v_{i,j}$ gives

$$\frac{-v_{i+1,j} + 2v_{i,j} - v_{i-1,j}}{h^2} + \frac{-v_{i,j+1} + 2v_{i,j} - v_{i,j-1}}{h^2} = f(x_i, y_j), \tag{IV.5}$$

with $v_{i,j} = 0$ along the boundary ($i = 0$ or $i = N$ or $j = 0$ or $j = N$) and $1 \le i \le N-1$ and $1 \le j \le N-1$. There are now $(N-1)^2$ interior grid points and unknowns. By arranging the lines of constant i in lexicographic order, the unknowns of the ith row of the grid can be collected in the vector $\mathbf{v_i} = (v_{i,1}, v_{i,2}, \ldots, v_{i,N-1})^T$ for $1 \le i \le N-1$. When $f(x_i, y_j)$ is ordered in the same manner, the system of equations (IV.5) can then be written in matrix form as

$$\frac{1}{h^2}\begin{bmatrix} A & -I & & & \\ -I & A & -I & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & -I \\ & & & & -I & A \end{bmatrix} \begin{bmatrix} \mathbf{v_1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{v_{N-1}} \end{bmatrix} = \begin{bmatrix} \mathbf{f_1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{f_{N-1}} \end{bmatrix}. \tag{IV.6}$$

This system is symmetric, block tridiagonal, and sparse. Each block on the diagonal is an $(N-1) \times (N-1)$ tridiagonal matrix, while each off-diagonal block is the negative of the $(N-1) \times (N-1)$ identity matrix I.

2. The Residual Equation

Remember that a system of linear equations, such as equations (IV.2) and (IV.5), can be written as

$$A\mathbf{u} = \mathbf{f}, \tag{IV.7}$$

with $\mathbf{u}$ being the exact solution. Let $\mathbf{v}$ represent an approximation to the exact solution. The exact solution is unknown, while the approximation $\mathbf{v}$, which is often computed using an iterative method, is known. One important measure of $\mathbf{v}$ as an approximation of $\mathbf{u}$ is algebraic error, given by

$$\mathbf{e} = \mathbf{u} - \mathbf{v}. \tag{IV.8}$$

However, since the exact solution is still not known, the error is also unknown. Still, one measure of how well $\mathbf{v}$ approximates $\mathbf{u}$ is the residual, given by

$$\mathbf{r} = \mathbf{f} - A\mathbf{v}. \tag{IV.9}$$

The residual is the amount by which the approximation fails to satisfy the original problem in (IV.7). Rewriting the residual, (IV.9), as

$$A\mathbf{v} = \mathbf{f} - \mathbf{r} \tag{IV.10}$$

and subtracting this new version from the original system, equation (IV.7), produces the residual equation

$$A\mathbf{e} = \mathbf{r}. \tag{IV.11}$$

Assuming that the approximation $\mathbf{v}$ has been computed by some method, the residual $\mathbf{r}$ can be easily computed using $\mathbf{r} = \mathbf{f} - A\mathbf{v}$. To improve the approximation, solve the residual equation (IV.11) for $\mathbf{e}$ and compute a new approximation using the definition of the error (IV.8). This is the idea of residual correction and will prove to be very important [Ref. 5].

The residual equation shows that the error satisfies the same set of equations as the desired exact solution when $\mathbf{f}$ is replaced by $\mathbf{r}$. This suggests that performing some iterative

method on the original equation $A\mathbf{v} = \mathbf{f}$ with an arbitrary guess $\mathbf{v}$ is equivalent to performing the same iterative method on the residual equation with the specific initial guess $\mathbf{e} = 0$.

3. Solving Systems of Linear Equations

There are several existing methods for solving sparse systems of linear equations like equations (IV.2) and (IV.4). *Direct methods* solve the system of equations in a known number of arithmetic operations. Several types of errors can result from using direct methods. Algebraic errors in the solution result from rounding errors introduced by the computation. Discretization error is the difference between the exact solution of the PDE and the exact solution of the difference equations being used to approximate the PDE. The magnitude of the discretization error at a grid point depends on the distances between consecutive discrete grid points and on the number of terms in the truncated series of differences used to approximate the derivatives.

Direct methods tend to be elimination methods, such as Gaussian elimination, or factorization methods, such as LU decomposition, QR factorization, or Singular Value Decomposition (SVD). As seen from the systems above, most difference approximations result in sparse matrices, with the number of zero entries greatly outnumbering the non-zero entries. For this type of matrix, a method like Gaussian elimination proves inefficient, because the procedure fills in the zero elements with non-zero elements during the elimination, thereby increasing the number of non-zero elements that need to be stored in the computer and used in later eliminations. As a result, direct methods can be quick and accurate for certain matrices with special properties but prove inefficient with elliptic equations [Ref. 3].

Iterative, or *relaxation*, methods are such that an initial approximation is used to calculate a second approximation which is used to calculate a third, and so on. Iterative methods are said to be convergent when the differences between successive approximations tends to zero as the number of iterations increases. Iterative methods have been much employed to solve $A\mathbf{u} = \mathbf{f}$ because they take advantage of the sparcity of A. These relaxation methods are easy to implement and can be successful in determining an exact solution for more general linear systems than the direct methods. However, experiments have shown that these methods generally work very well only for the first several iterations. Convergence then slows and the method appears to stall.

Iterative methods, such as the Jacobi and Gauss-Seidel methods, begin with an initial guess. Valuable insight can be gained by using an initial guess consisting of Fourier modes:

$$v_j = sin\left(\frac{jk\pi}{N}\right), \quad 0 \leq j \leq N, \quad 1 \leq k \leq N-1. \tag{IV.12}$$

The integer k is the *wavenumber* and indicates the number of half sine waves which make up the vector $\mathbf{v}$ on the domain. Small values of k correspond to long, smooth waves. Large values correspond to oscillatory waves. The rapid decrease in error during the first several applications of an iterative method results from efficient elimination of the oscillatory modes of this error. Unfortunately, the iteration is much less effective in reducing any remaining smooth components. Most relaxation methods possess this property, known as the *smoothing property* [Ref. 5]. The issue is whether basic iterative methods can be adapted to make them effective on all error components.

One way to improve an iterative method is to start with a good initial guess. By performing some preliminary iterations on a coarse grid, an improved initial guess can be

derived and used on the original fine grid. This coarse grid is a grid (Figure 14) with a larger interval between grid points. Performing these iterations, or relaxing, on a coarse grid is less expensive since there are fewer unknowns to be updated.

With this is mind, recall that most relaxation schemes are effective at eliminating the oscillatory components of error but prove futile against the smooth components. Now, consider these remaining smooth components on a coarser grid. A smooth curve on a fine grid can appear more oscillatory on a coarse grid. Note in Figure 15 that the grid points on the coarse grid (bottom grid) correspond to the even-numbered grid points of the fine grid (top grid). Consider the kth mode of the fine grid evaluated at the even-numbered grid points. If $1 \leq k \leq N/2$, then the components of the kth mode on the fine grid can be written as

$$w^h_{k,2j} = sin\left(\frac{2jk\pi}{N}\right) = sin\left(\frac{jk\pi}{N/2}\right) = w^{2h}_{k,j}. \tag{IV.13}$$

The superscripts on $\mathbf{w}$ indicate on which grid the vector is defined by giving the grid spacing on the grid. The vector $\mathbf{w}^h$ is then defined on the fine grid, while the vector $\mathbf{w}^{2h}$ is defined on the coarse grid.

Thus, the kth mode on the fine grid becomes the kth mode on the coarse grid. The kth mode also becomes more oscillatory in passing from the fine grid to the coarse grid. From figure 15, $k = 4$ on the grid where $N = 12$ is $1/3$ of the highest possible frequency, while $k = 4$ on the grid where $N = 6$ is $2/3$ of the highest possible, making the wave more oscillatory on the coarse grid. This is true only for $1 \leq k \leq N/2$. If $k = N/2$, the kth mode on the fine grid is the zero vector on the coarse grid. And for $k > N/2$, the kth mode on the fine grid becomes the $(N - k)$th mode on the coarse grid. When this phenomenon, known

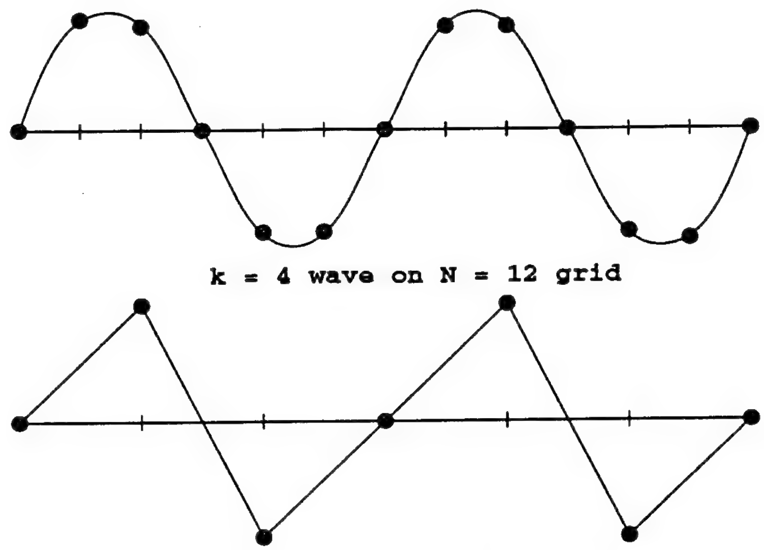

Figure 15. *A wave with wavenumber* $k = 4$ *on a fine grid* $(N = 12)$ *appears more oscillatory on a coarse grid* $(N = 6)$.

as *aliasing* , occurs, oscillatory modes on the fine grid are misrepresented as smooth modes on the coarse grid.

All of this suggests that, once convergence slows on the fine grid, the residual may be computed and the residual equation can be relaxed on a coarser grid to obtain an approximation of the error. This error can then be returned to the fine grid to correct the original approximation. This method for improving iterative approximations, called *coarse grid correction*, leaves some issues unresolved, such as how to transfer vectors from a coarse grid to a fine grid (and back again), how to compute the residual on the fine grid, and how to compute an error estimate on the coarse grid.

4. Intergrid Transfers

Before an iterative method can be used on coarser grids, a mechanism is needed for tranferring vectors between grids. A discussion of intergrid transfers will deal only with the

case in which the coarse grid has twice the grid spacing of the next finest grid, since there seems to be no advantage in using grid spacing with ratios other than two.

Transferring a vector from a coarse grid to a fine grid is called *interpolation*, or *prolongation*. Although many methods exist, for most multigrid purposes linear interpolation is quite effective. The linear interpolation operator will be denoted as I_{2h}^h. The operator takes coarse grid vectors and produces fine grid vectors. With $\mathbf{v}^{2h}$ as the coarse grid vector and $\mathbf{v}^h$ as the fine grid vector, the notation $I_{2h}^h \mathbf{v}^{2h} = \mathbf{v}^h$ transfers the vectors such that

$$\left.\begin{array}{rcl} v_{2j}^h & = & v_j^{2h} \\ v_{2j+1}^h & = & \frac{1}{2}\left(v_{j+1}^{2h} + v_j^{2h}\right) \end{array}\right\} \; 0 \leq j \leq \frac{N}{2} - 1.$$

This transfer notation shows that, at even-numbered fine grid points, the values of the vector are transferred directly from the coarse to the fine grid. At odd-numbered fine grid points, the value of the grid point on the fine grid is the average of the adjacent coarse grid values.

For a two-dimensional problem, the interpolation operator is defined in a similar manner. According to the notation $I_{2h}^h \mathbf{v}^{2h} = \mathbf{v}^h$, the components of $\mathbf{v}^h$ are given by

$$\left.\begin{array}{rcl} v_{2i,2j}^h & = & v_{i,j}^{2h} \\ v_{2i+1,2j}^h & = & \frac{1}{2}(v_{i+1,j}^{2h} + v_{i,j}^{2h}) \\ v_{2i,2j+1}^h & = & \frac{1}{2}(v_{i,j+1}^{2h} + v_{i,j}^{2h}) \\ v_{2i+1,2j+1}^h & = & \frac{1}{4}(v_{i+1,j+1}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i,j}^{2h}) \end{array}\right\} \; 0 \leq i,j \leq \frac{N}{2} - 1.$$

Transferring a vector from a fine to a coarse grid is called *restriction*. One restriction operator, *injection*, is defined by $I_h^{2h}\mathbf{v}^h = \mathbf{v}^{2h}$, where the components of $\mathbf{v}^{2h}$ are given by

$$v_j^{2h} = v_{2j}^h.$$

With injection, the coarse grid vector takes its value directly from the corresponding fine grid point.

Another restriction operator, known as *full weighting*, is defined by $I_h^{2h}\mathbf{v}^h = \mathbf{v}^{2h}$, where the values of the coarse grid vector are a weighted average of values at neighboring fine grid points, or

$$v_j^{2h} = \frac{1}{4}\left(v_{2j+1}^h + 2v_{2j}^h + v_{2j-1}^h\right), \qquad 1 \le j \le \frac{N}{2} - 1. \tag{IV.14}$$

The two-dimensional full weighting operator is defined similarly, where the values of the coarse grid vector take on a weighted average of values at neighboring fine grid points. Thus,

$$\begin{aligned} v_{i,j}^{2h} &= \tfrac{1}{16}\Big[v_{2i-1,2j-1}^h + v_{2i-1,2j+1}^h + v_{2i+1,2j-1}^h + v_{2i+1,2j+1}^h \\ &\quad +2\left(v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h\right) \\ &\quad +4v_{2i,2j}^h\Big], \qquad 1 \le i,j \le \tfrac{N}{2} - 1. \end{aligned}$$

5. Coarse Grid Correction

A coarse grid correction scheme may now be introduced. Coarse-grid correction is described as a two-grid process as follows:

- Relax ν_1 times on $A^h\mathbf{u}^h = \mathbf{f}$ on Ω^h with an initial approximation $\mathbf{v}^h$.
- Compute the residual on the fine grid, $\mathbf{r}^h = \mathbf{f}^h - A\mathbf{v}^h$, transfer the residual from the fine grid to the coarse grid, $\mathbf{r}^{2h} = I_h^{2h}\mathbf{r}^h$, and solve the residual equation, $A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}$, on the coarse grid Ω^{2h} with an initial guess $\mathbf{e}^{2h} = 0$.
- Correct the approximation obtained on Ω^h ($\mathbf{v}^h$ in the first step) with the error estimate obtained on Ω^{2h}: $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h\mathbf{e}^{2h}$.
- Relax ν_2 times on $A^h\mathbf{u}^h = \mathbf{f}$ on Ω^h with the improved initial approximation $\mathbf{v}^h$.

The integers ν_1 and ν_2 are parameters in the correction scheme which control the number of relaxation sweeps before and after the coarse grid correction. Experiments have shown that effective choices for ν_1 and ν_2 are often 1, 2, or 3.

A^{2h} is the coarse grid representation of the original matrix A. The original problem could be directly discretized on Ω^{2h} using the same techniques used to generate A^h, the fine grid operator, but at a spacing of $2h$, resulting in A^{2h}. An alternate approach is to note that if the error on the fine grid, $\mathbf{e}^h = \mathbf{u}^h - \mathbf{v}^h$, lies entirely within the range of interpolation, denoted by $R(I_{2h}^h)$, then the error can be represented by $\mathbf{e}^h = I_{2h}^h \mathbf{u}^{2h}$, for some coarse grid vector $\mathbf{u}^{2h}$. The residual equation on Ω^h can then be written as

$$A^h \mathbf{e}^h = A^h I_{2h}^h \mathbf{u}^{2h} = \mathbf{r}^h.$$

Applying the restriction operator I_h^{2h} to the equation

$$A^h I_{2h}^h \mathbf{u}^{2h} = \mathbf{r}^h$$

gives

$$I_h^{2h} A^h I_{2h}^h \mathbf{u}^{2h} = I_h^{2h} \mathbf{r}^h$$

which leads to the natural definition

$$A^{2h} = I_h^{2h} A^h I_{2h}^h.$$

The coarse grid equation becomes

$$A^{2h} \mathbf{u}^{2h} = \mathbf{r}^{2h}.$$

This argument is based on the assumption that the error $\mathbf{e}^h$ is entirely within the range of interpolation. This may not always be the case. If the error were entirely within this range

of interpolation, then solving the residual equation on the coarse grid would give an exact solution. Regardless, this approach gives a definition for the coarse grid operator A^{2h}. This definition is known as the *Galerkin condition.* This coarse grid operator A^{2h} may not give the same A^{2h} as discretizing at twice the mesh spacing. However, the Galerkin condition may be used to prove such things as convergence theorems, making this condition an effective theoretical tool.

In the above coarse grid scheme, there is an efficient way of solving the coarse grid problem $A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}$. The coarse grid problem is not much different from the original problem, and can be solved by the same coarse grid correction scheme. Applying the coarse grid correction scheme to the residual equation on the coarse grid requires a move to the next coarsest grid, Ω^{4h}, for the correction step. This process is repeated on successively coarser grids until a direct solution to the residual equation is economical. The resulting algorithm is applied recursively down to the coarsest grid, sometimes a single interior grid point. Corrections are then applied successively through finer and finer grids, returning eventually to the finest grid. This algorithm is referred to as a *V-cycle* , named for the pictoral representation of the schedule of grid visits (see Figure 16). It is defined by $\mathbf{v}^h \leftarrow MV^h(\mathbf{v}^h, \mathbf{f}^h)$ and is summarized here using a compact recursive definition:

•Relax ν_1 times on $A^h\mathbf{u}^h = \mathbf{f}$ on Ω^h with an initial approximation $\mathbf{v}^h$.

•If Ω^h is the coarsest grid, then proceed to the last step. Otherwise, compute the residual on the fine grid, $\mathbf{r}^h = \mathbf{f}^h - A\mathbf{v}^h$, transfer the residual from the fine grid to the next coarsest grid, $\mathbf{r}^{2h} = I_h^{2h}\mathbf{r}^h$, and set the initial guess for Ω^{2h} to zero, $\mathbf{v}^{2h} \leftarrow 0$. Return to the first step, $\mathbf{v}^{2h} \leftarrow MV^{2h}(\mathbf{v}^{2h}, \mathbf{r}^{2h})$.

•Correct $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h\mathbf{v}^{2h}$

•Relax ν_2 times on the original equation, $A^h\mathbf{u}^h = \mathbf{f}^h$, with an initial guess $\mathbf{v}^h$.

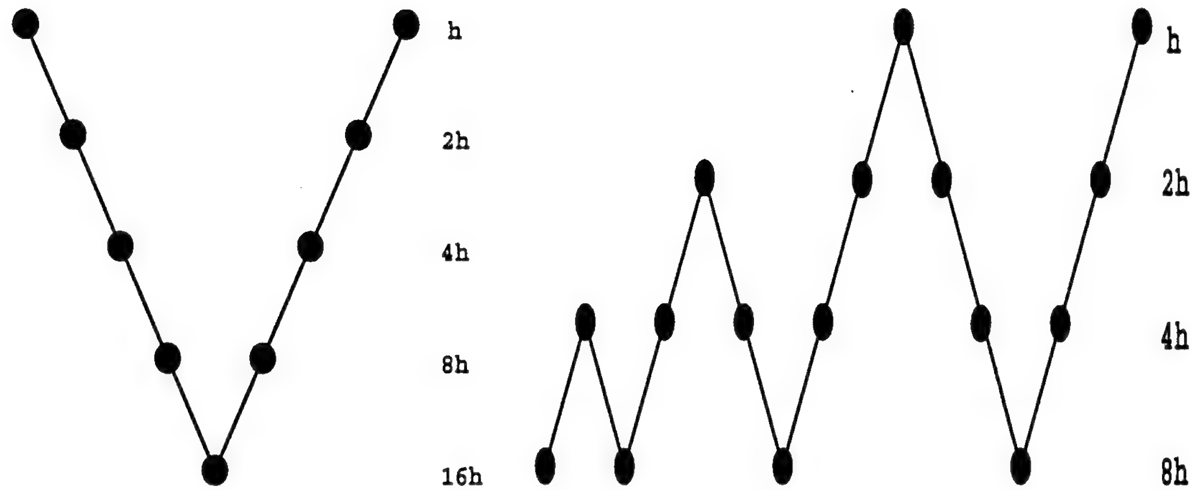

Figure 16. *Schedule of grids for the V-cycle and the FMV scheme.*

Recall that one way to improve iterative methods is to start with a good initial guess. The V-cycle uses only coarse grid correction in solving the residual equation and does not make use of any method to improve the initial guess. The method of *nested iteration* uses coarse grids to obtain improved initial guesses for fine grid problems. In order to obtain an improved initial guess for the first fine grid relaxation of the V-cycle, nested iteration suggests solving this problem on the next coarsest grid. To obtain an improved initial guess on this coarse grid, say Ω^{2h}, it is necessary to solve the problem on Ω^{4h}. This pattern results in a recursive path that leads to the coarsest grid.

The full multigrid V-cycle (FMV) joins nested iteration with the V-cycle, resulting in a simple but powerful algorithm. Each V-cycle is preceded by a smaller V-cycle that provides the best initial guess possible. The full multigrid V-cycle, defined by $\mathbf{v}^h \leftarrow FMV^h(\mathbf{v}^h, \mathbf{f}^h)$, is summarized here:

•Initialize $\mathbf{f}^h$, $\mathbf{r}^{2h}$, $\mathbf{r}^{4h}$, ...; $\mathbf{v}^h$, $\mathbf{e}^{2h}$, $\mathbf{e}^{4h}$, ... to zero.

•Relax on the residual equation on the coarsest grid (or solve directly if the coarsest grid is small).

•

•

•

•Correct $\mathbf{e}^{4h} \leftarrow \mathbf{e}^{4h} + I_{8h}^{4h}\mathbf{e}^{8h}$ and perform a V-cycle using the grid Ω^{4h} as the finest grid and Ω^{8h} as the coarsest grid. This is abbreviated as $\mathbf{e}^{4h} \leftarrow MV^{4h}(\mathbf{e}^{4h}, \mathbf{r}^{4h})$.

•Correct $\mathbf{e}^{2h}$ using the result from the previous V-cycle, $\mathbf{e}^{2h} \leftarrow \mathbf{e}^{2h} + I_{4h}^{2h}\mathbf{e}^{4h}$.

•$\mathbf{e}^{2h} \leftarrow MV^{2h}(\mathbf{e}^{2h}, \mathbf{r}^{2h})$.

•$\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^{h}\mathbf{v}^{2h}$.

•$\mathbf{v}^h \leftarrow MV^h(\mathbf{v}^h, \mathbf{f}^h)$.

The last step in the above procedure is the original V-cycle discussed earlier, but is preceded by several smaller V-cycles (Figure 16) in order to make an improved initial guess.

The full multigrid V-cycle brings together many ideas and techniques, that, when taken alone, can have serious limitations. Full multigrid integrates these ideas so that they may work together to remove these limitations.

B. THE FAST ADAPTIVE COMPOSITE GRID METHOD

The fast adaptive composite grid method (FAC) is a natural extension of conventional multigrid methods applied to problems discretized on composite grids. Multigrid was developed as a global grid solver but is not efficient for cases where fine grid resolution and accuracy are needed only in a local subdomain. While a certain amount of accuracy may be suitable to correctly solve the physical system in most of the global domain, greater accuracy is most often needed in a given local region. Obtaining the desired accuracy may take a resolution with a mesh size h, which is smaller than that in the rest of the global domain.

If a global fine grid were incorporated, unnecessary computations would result outside the local region, often at a great computational cost [Ref. 4].

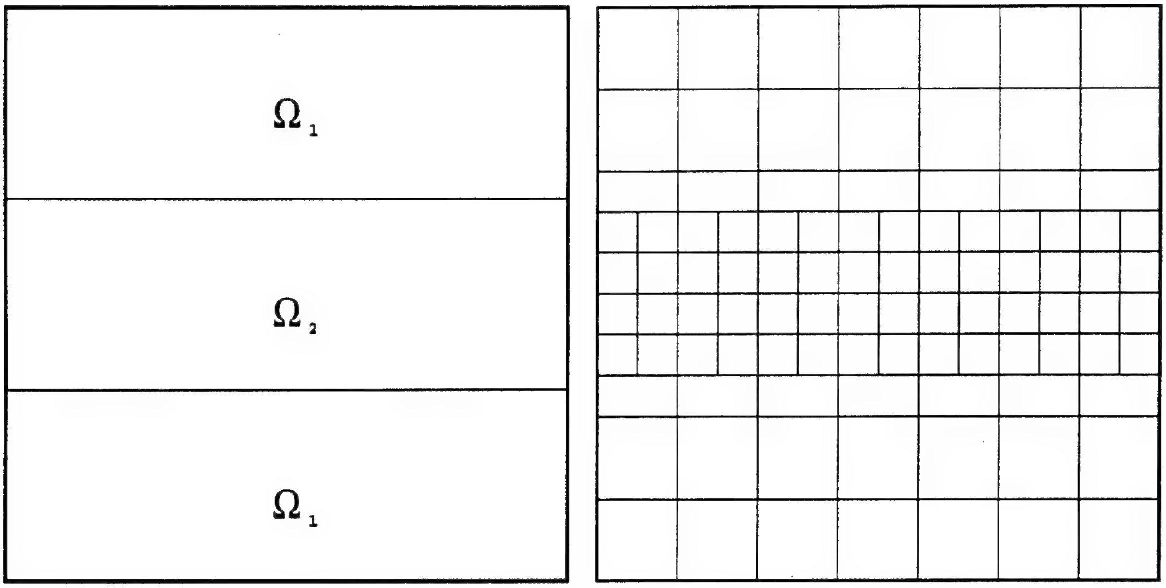

Figure 17. *A global domain Ω. Ω_2 is the local domain with a smaller grid spacing while Ω_1 is that portion of the global domain Ω that retains the original coarse grid spacing.*

To arrive at FAC, the desired method, ideas will be drawn from several simpler, less efficient methods. It should be noted here that this development will follow closely with that of reference [Ref. 4].

1. Local Multigrid

A natural way to avoid using a global fine grid without sacrificing the necessary fine grid resolution in the local subdomain is to modify the standard multigrid scheme by simply suppressing relaxation outside of the subdomain. Let $G^h_{\Omega_2}$ represent a restricted relaxation process on the local subdomain Ω_2 (Figure 17). Let the decomposition of $\mathbf{u}^h$ be represented by

$$\mathbf{u}^h = \begin{pmatrix} \mathbf{u}^h_1 \\ \mathbf{u}^h_2 \end{pmatrix},$$

where $\mathbf{u}_2^h$ represents the unknown values of $\mathbf{u}$ found on the local subdomain, Ω_2, while the remaining unknown values, $\mathbf{u}_1^h$, are found on the complement of the subdomain, $C\Omega_2$. Applying a similar decomposition to $\mathbf{f}^h$, the restricted relaxation operator has the form

$$G_{\Omega_2}^h(\mathbf{u}_2^h, \mathbf{f}_2^h) = \begin{pmatrix} \mathbf{u}_1^h \\ G^h(\mathbf{u}^h, \mathbf{f}^h) \end{pmatrix},$$

where G^h is a relaxation scheme corresponding to the local problem on Ω_2, resulting in changes to $\mathbf{u}_2^h$ only. The *local multigrid method* , defined by $\mathbf{u}^h \leftarrow MG_{\Omega_2}^h(\mathbf{u}^h, \mathbf{f}^h)$, can then be expressed in the two-grid form as follows:

- $\mathbf{f}^{2h} \leftarrow I_h^{2h}(\mathbf{f}^h - A^h\mathbf{u}^h)$
- $\mathbf{u}^{2h} \leftarrow (A^{2h})^{-1}\mathbf{f}^{2h}$
- $\mathbf{u}^h \leftarrow \mathbf{u}^h + I_{2h}^h\mathbf{u}^{2h}$
- $\mathbf{u}^h \leftarrow G_{\Omega_2}^h(\mathbf{u}^h, \mathbf{f}^h)$

The objective of the local multigrid scheme $MG_{\Omega_2}^h$ is to eliminate unnecessary computations on $C\Omega_2$. But only the relaxation process has been suppressed there, not the intergrid transfers or the residual calculations. Fortunately, the intergrid transfers and residual calculations can be suppressed without changing the final result. With $\mathbf{u}^h$ as the initial approximation for one $MG_{\Omega_2}^h$ cycle, $\mathbf{u}^{2h}$ as the correction computed in the above two-grid form, and $\mathbf{v}^h$ as the change in $\mathbf{u}^h + I_{2h}^h\mathbf{u}^{2h}$ resulting from the relaxation on Ω_2^h (again from above), the approximation produced by $MG_{\Omega_2}^h$ is $\mathbf{u}^h + I_{2h}^h\mathbf{u}^{2h} + \mathbf{v}^h$. Since $\mathbf{v}_h$ is zero on $C\Omega_2$ (no relaxation occurs on $C\Omega_2$ in the relaxation scheme), $I_{2h}^h A_h \mathbf{v}^h$ is zero on $C\bar{\Omega}_2^h$, or rather on the set of points that are not in Ω_2^h or its interface ($\Omega_1^h \bigcap C\bar{\Omega}_2$). Using the two-grid form above and

the Galerkin condition $I_h^{2h} A^h I_{2h}^h = A^{2h}$ results in

$$\begin{aligned} I_h^{2h}(\mathbf{f}^h - A^h(\mathbf{u}^h + I_{2h}^h\mathbf{u}^{2h} + \mathbf{v}^h)) &= I_h^{2h}(\mathbf{f}^h - A^h\mathbf{u}^h) - I_h^{2h}A^hI_{2h}^h\mathbf{u}^{2h} - I_h^{2h}A^h\mathbf{v}^h \\ &= I_h^{2h}(\mathbf{f}^h - A^h\mathbf{u}^h) - A^{2h}\mathbf{u}^{2h} - I_h^{2h}A^h\mathbf{v}^h \\ &= \mathbf{f}^{2h} - \mathbf{f}^{2h} - I_h^{2h}A^h\mathbf{v}^h \\ &= -I_h^{2h}A^h\mathbf{v}^h. \end{aligned}$$

This implies that the transferred residual $\mathbf{f}^{2h}$ in the next cycle of $MG_{\Omega_2}^h$ will be zero at the points of $C\bar{\Omega}_2^{\mathrm{h}}$. In general, local fine grid relaxation has no immediate effect on the coarse grid equations at the points on $C\bar{\Omega}_2^{\mathrm{h}}$. There is no need to use the fine grid in that region [Ref. 4].

2. Bordered Multilevel Method

To prevent unnecessary work away from the local fine grid, the role of $\mathbf{u}^{2h}$ in $C\bar{\Omega}_2^{\mathrm{h}}$ must change from representing error corrections to representing the current approximation. A modification of the local multigrid scheme $MG_{\Omega_2}^h$ uses this $\mathbf{u}^{2h}$ to accumulate corrections. Also, when using this modification of $MG_{\Omega_2}^h$, the fine grid must have enough data to compute $I_h^{2h}(\mathbf{f}^h - A^h\mathbf{u}^h)$ at the coarse grid interface points. This requires interpolation on $\bar{\Omega}_2^h$ and on the two grid lines bordering $\bar{\Omega}_2^h$. Figure 18 shows the first border and the extended local fine grid, $\bar{\Omega}_{2+}^h$, which includes both borders.

These ideas provide for a new algorithm, called the *bordered mulilevel* scheme, that avoids all fine grid work away from the local fine grid Ω_2^h. In this new algorithm, residuals are computed on the extended local fine grid, which includes the local fine grid, the boundary points of the local fine grid, and the first border. The residuals are transferred to the

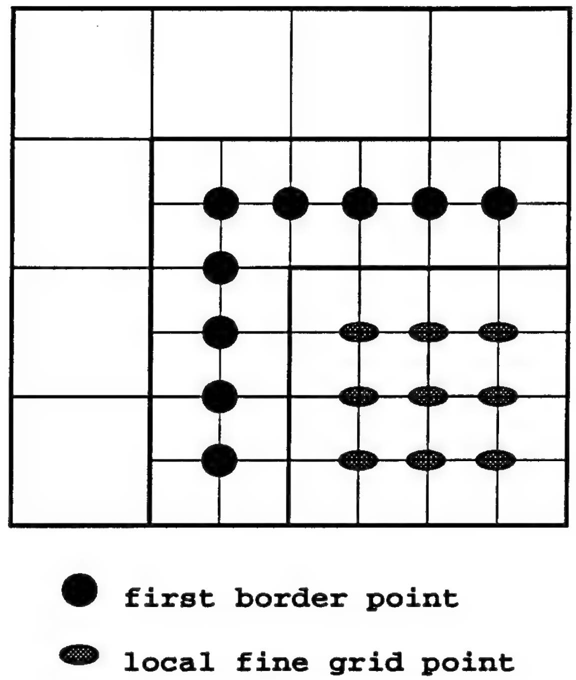

Figure 18. *The local grid $\bar{\Omega}_2^h$, its first border (indicated by $\bullet$), and the extended local grid $\bar{\Omega}_{2+}^h$.*

coarse grid points lying under the fine grid and the interface (residuals are also computed on the coarse grid outside the fine grid). A global coarse-grid correction is computed and interpolated to the extended local fine grid. Relaxation is then performed at fine grid interior points.

3. Multilevel Composite Grid Method

The bordered multilevel scheme requires two additional grid lines bordering the local fine grid in order to compute the residual correction at the coarse grid interface points. This residual correction is the composite grid residual at the coarse grid interface points. Let $\Omega^{\underline{h}}$ denote the composite grid, $\Omega^{\underline{h}} = \Omega^{2h} \bigcup \Omega^h$. By generating a composite grid operator, $A^{\underline{h}} : U^{\underline{h}} \rightarrow U^{\underline{h}}$ where $U^{\underline{h}}$ is the nodal space associated with the composite grid $\Omega^{\underline{h}}$, the residual correction can be computed more directly without the need for any borders. This

leads to the *multilevel composite grid method*, represented by $\mathbf{u}^{\underline{h}} \leftarrow ML^{\underline{h}}(\mathbf{u}^{\underline{h}}, \mathbf{f}^{\underline{h}})$, and defined as follows:

- •Set $\mathbf{f}^{2h} = I_{\underline{h}}^{2h}(\mathbf{f}^{\underline{h}} - A^{\underline{h}}\mathbf{u}^{\underline{h}})$.
- •Compute $\mathbf{u}^{2h} \leftarrow (A^{2h})^{-1}\mathbf{f}^{2h}$.
- •Compute $\mathbf{u}^{\underline{h}} \leftarrow \mathbf{u}^{\underline{h}} + I_{2h}^{\underline{h}}\mathbf{u}^{2h}$.
- •Set $\mathbf{f}^{h} = I_{\underline{h}}^{h}(\mathbf{f}^{\underline{h}} - A^{\underline{h}}\mathbf{u}^{\underline{h}})$.
- •Set $\mathbf{u}^{h} \leftarrow 0$ (the initial guess for the fine grid is zero) and compute $\mathbf{u}^{h} \leftarrow G^{h}(\mathbf{u}^{h}, \mathbf{f}^{h})$.
- •Compute $\mathbf{u}^{\underline{h}} \leftarrow \mathbf{u}^{\underline{h}} + I_{h}^{\underline{h}}\mathbf{u}^{h}$.

In developing this multilevel composite scheme, it has become necessary to define some new notation. First, the domain Ω needs to be partitioned as

$$\Omega = \Omega_C \bigcup \Omega_I \bigcup \Omega_F \tag{IV.15}$$

which breaks the domain into subdomains. Ω_C is that part of the domain that contains all coarse grid points excluding the interface points, Ω_I includes the interface points, and Ω_F includes the fine grid points. On the composite grid in Figure 19, the grid $\Omega^{\underline{h}}$ is partitioned into the following:

$$\Omega^{\underline{h}} = \Omega_C^{\underline{h}} \bigcup \Omega_I^{\underline{h}} \bigcup \Omega_F^{\underline{h}}. \tag{IV.16}$$

$\Omega_C^{\underline{h}}$ consists of the coarse grid points (excluding the interface points), $\Omega_I^{\underline{h}}$ includes the coarse grid interface points, and $\Omega_F^{\underline{h}}$ is the fine grid patch.

A block decomposition of the operators and grid equations results from the partition in (IV.16). The block decomposition of the grid equations $A^{\underline{h}}\mathbf{u}^{\underline{h}} = \mathbf{f}^{\underline{h}}$ is given by

$$\begin{pmatrix} A_{CC}^{\underline{h}} & A_{CI}^{\underline{h}} & A_{CF}^{\underline{h}} \\ A_{IC}^{\underline{h}} & A_{II}^{\underline{h}} & A_{IF}^{\underline{h}} \\ A_{FC}^{\underline{h}} & A_{FI}^{\underline{h}} & A_{FF}^{\underline{h}} \end{pmatrix} \begin{pmatrix} \mathbf{u}_C^{\underline{h}} \\ \mathbf{u}_I^{\underline{h}} \\ \mathbf{u}_F^{\underline{h}} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_C^{\underline{h}} \\ \mathbf{f}_I^{\underline{h}} \\ \mathbf{f}_F^{\underline{h}} \end{pmatrix}.$$

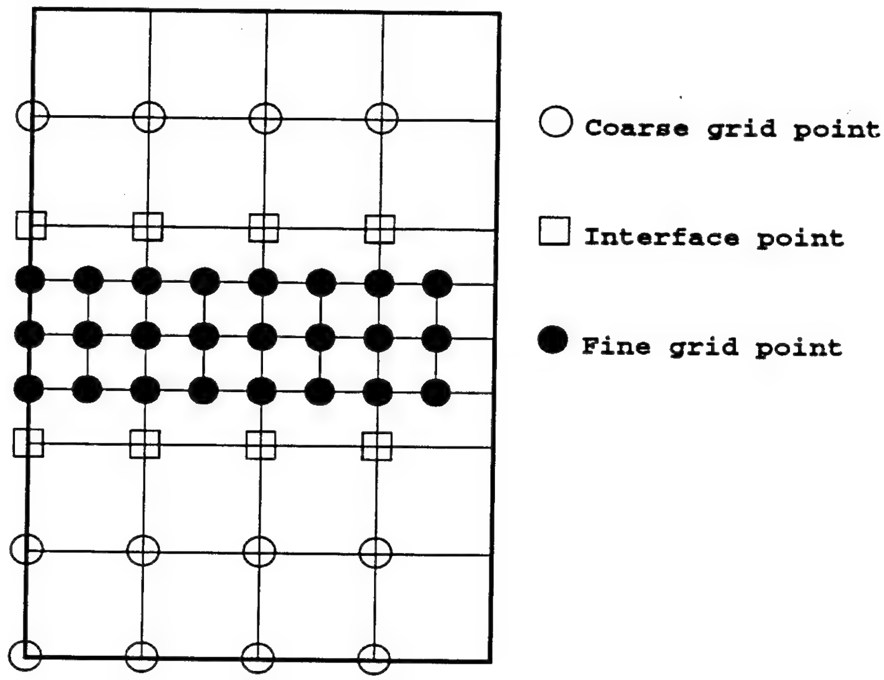

Figure 19. *Composite grid partition. Subgrid Ω_C^h contains only coarse grid points, subgrid Ω_I^h contains only interface points, and subgrid Ω_F^h contains only fine grid points.*

The notation A_{XY}^h signifies that the equation being solved corresponds to an X-grid point and the stencil reaches to a Y-grid point. If $\bar{I}_{2h}^h$ and $\bar{I}_h^{2h}$ represent the grid transfer operators restricted to the local fine grid, then certain observations result from the way the grid operators are constructed. For the transfer equation $\mathbf{u}^h = I_{2h}^h \mathbf{u}^{2h}$, the transfer operator is

$$I_{2h}^h = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \star & \bar{I}_{2h}^h \end{pmatrix}, \tag{IV.17}$$

which may be interpreted as follows: outside the local fine patch, interpolation is simply the identity. Interpolating $\mathbf{u}^{2h}$ inside the local fine patch is accomplished using standard prolongation away from the borders. The star, $\star$, represents actual storage of the slave points as Ω_C^h values.

The transfer operator $I_{\underline{h}}^{2h}$ is given by

$$I_{\underline{h}}^{2h} = \begin{pmatrix} I & 0 & 0 \\ 0 & I & \star \\ 0 & 0 & \bar{I}_h^{2h} \end{pmatrix}. \tag{IV.18}$$

Residuals outside the local fine patch are transferred by injection to correct the equations on the coarse grid $(\Omega_C^{2h} \bigcup \Omega_I^{2h})$. These residuals do not affect Ω_F^{2h} equations. Rather, residuals in $\Omega_F^{\underline{h}}$ are used to correct Ω_F^{2h} equations. Residuals in $\Omega_I^{\underline{h}} \bigcup \Omega_F^{\underline{h}}$ are used to correct Ω_I^{2h} equations.

The transfer operator $I_{\underline{h}}^{h}$ is given by

$$I_{\underline{h}}^{h} = \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}. \tag{IV.19}$$

Residuals are transferred from the composite grid to the fine grid patch in the trivial manner.

The transfer operator $I_h^{\underline{h}}$ is given by

$$I_{\underline{h}}^{h} = \begin{pmatrix} 0 & 0 & I \end{pmatrix}. \tag{IV.20}$$

Interpolation from the fine grid patch to the composite grid is the trivial interpolation, the identity.

The composite grid operator is given, as before, by

$$A^{\underline{h}} = \begin{pmatrix} A_{CC}^{\underline{h}} & A_{CI}^{\underline{h}} & A_{CF}^{\underline{h}} \\ A_{IC}^{\underline{h}} & A_{II}^{\underline{h}} & A_{IF}^{\underline{h}} \\ A_{FC}^{\underline{h}} & A_{FI}^{\underline{h}} & A_{FF}^{\underline{h}} \end{pmatrix}. \tag{IV.21}$$

Recall that the notation $A_{XY}^{\underline{h}}$ means that the equation being solved corresponds to an X-grid point and the stencil reaching to a Y-grid point. Since the coarse grid stencils do not reach any fine grid points, $A_{FC}^{\underline{h}}$ and $A_{CF}^{\underline{h}}$ are zero.

The composite grid operator can then be written as

$$A^{\underline{h}} = \begin{pmatrix} A_{CC}^{2h} & \star & 0 \\ \star & \star & \star \\ 0 & \star & A^h \end{pmatrix}. \tag{IV.22}$$

The composite grid operator is block tridiagonal and agrees with the coarse grid operator on Ω_C and with the fine grid operator on Ω_F. The stars, $\star$, are non-zero but unspecified entries whose values are not important at this time.

The composite grid residuals in Ω_C and Ω_F can be computed using the respective coarse and fine grid stencils. Special evaluation is, however, required at the coarse grid interface points. The stencils for $A^{\underline{h}}$ in the multilevel composite grid scheme $ML^{\underline{h}}$ must be computed for use on $\Omega_I^{\underline{h}}$.

The multilevel composite grid scheme $ML^{\underline{h}}$ given above is written in the *immediate correction* form. This scheme is considered to be "immediate correction" because both intermediate quantities $\mathbf{u}^{2h}$ and $\mathbf{u}^h$ attempt to approximate the current algebraic error $\mathbf{e}^{\underline{h}} = \mathbf{u}^{\underline{h}^*} - \mathbf{u}^{\underline{h}}$ ($\mathbf{u}^{\underline{h}^*}$ is the exact solution) while solving the residual equation

$$A^{\underline{h}}\mathbf{e}^{\underline{h}} = \mathbf{f}^{\underline{h}} - A^{\underline{h}}\mathbf{u}^{\underline{h}}. \tag{IV.23}$$

This immediate correction form attempts to solve the residual equation (IV.23) by transferring the composite grid residual to each level in turn, computing a correction there, and immediately interpolating it back to the composite grid.

4. The Fast Adaptive Composite Method

A final modification to the multilevel composite grid scheme $ML^{\underline{h}}$ is made by replacing fine grid relaxation with a direct solver. The replacement of the fine grid relaxation with a

direct solver is given by

$$G^h(\mathbf{u}^h, \mathbf{f}^h) = (L^h)^{-1}\mathbf{f}^h.$$

This modification gives the *fast adaptive composite grid* method (FAC). This method is represented by $\mathbf{u}^{\underline{h}} \leftarrow FAC^{\underline{h}}(\mathbf{u}^{\underline{h}}, \mathbf{f}^{\underline{h}})$ and is defined as follows:

- •Set $\mathbf{f}^{2h} = I_{\underline{h}}^{2h}(\mathbf{f}^{\underline{h}} - A^{\underline{h}}\mathbf{u}^{\underline{h}})$.
- •Compute $\mathbf{u}^{2h} \leftarrow (A^{2h})^{-1}\mathbf{f}^{2h}$.
- •Compute $\mathbf{u}^{\underline{h}} \leftarrow \mathbf{u}^{\underline{h}} + I_{2h}^{\underline{h}}\mathbf{u}^{2h}$.
- •Set $\mathbf{f}^{h} = I_{\underline{h}}^{h}(\mathbf{f}^{\underline{h}} - A^{\underline{h}}\mathbf{u}^{\underline{h}})$.
- •Compute $\mathbf{u}^{h} \leftarrow (L^{h})^{-1}\mathbf{f}^{h}$.
- •Compute $\mathbf{u}^{\underline{h}} \leftarrow \mathbf{u}^{\underline{h}} + I_{h}^{\underline{h}}\mathbf{u}^{h}$.

C. DISCRETIZATION ON THE COMPOSITE GRID

To apply FAC, the equations to be solved must be discretized on a composite grid. FVE provides the method for this discretization since FVE has proven effective with PDEs on a composite grid.

1. Discretization of the Model Problem on a Composite Grid

As an example, recall that the model problem is the potential equation

$$\begin{aligned} \nabla \cdot (\rho \nabla \psi) &= \eta \quad \text{in } \Omega \qquad \text{(IV.24)} \\ \psi &= \psi_0 \quad \text{on } \partial\Omega_N \bigcup \partial\Omega_E \text{ (Dirichlet)} \\ (\rho \nabla \psi) \cdot \hat{n} &= \psi_1 \quad \text{on } \partial\Omega_W \bigcup \partial\Omega_S \text{ (Neumann).} \end{aligned}$$

The composite grid can be taken as a coarse grid with a patch of fine grid points, where the coarse grid points and the slave interface points along the boundary of the fine grid patch

appear in the patch equations as if they were Dirichlet boundary points for the patch [Ref. 4]. The patch volumes are chosen first, with the volumes along the interface being largely dictated by the neighboring patch volumes. Triangulation follows with the patch geometry being determined at the interface by combining pairs of triangles of opposing orientation that abut the slave points (Figure 20).

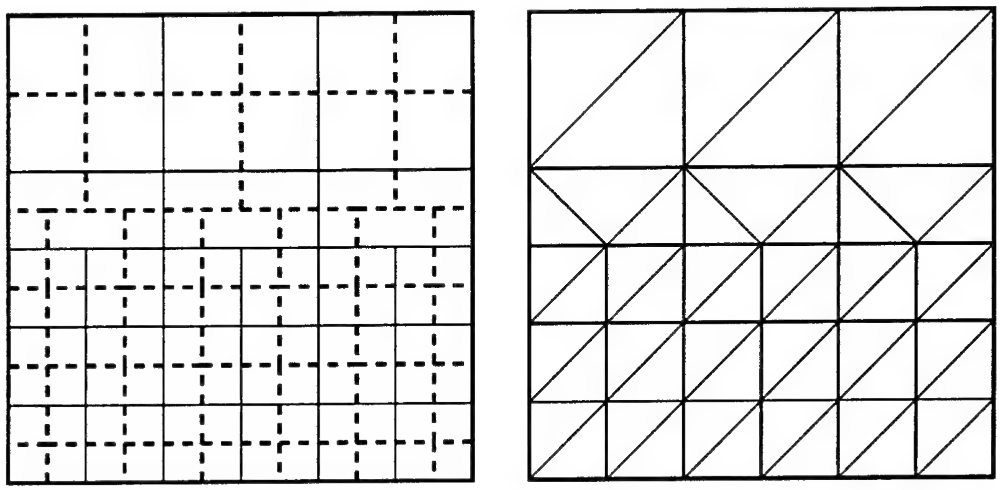

Figure 20. *Example of composite grid volumes and composite grid triangulation.*

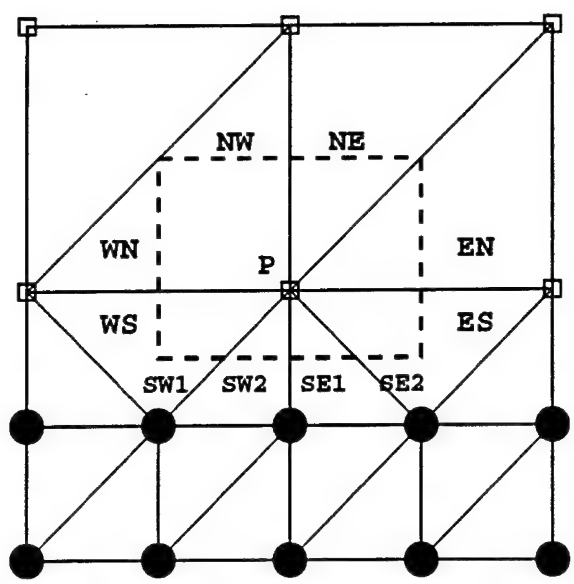

Figure 21. *Interface point volume for a composite grid.*

As an example of a stencil for a composite grid, consider the interface point P in Figure 21. The control volume is smaller for the interface point than a control volume for the same point on a uniform coarse grid, and there are now seven triangular regions over which to integrate. In the seven regions are ten portions of the border of the volume. Using the same procedures as for the general interior grid point of the uniform grid, integrate over each of the ten segments of the volume boundary. The resulting stencil form of the equation for the interface point P is

$$\begin{bmatrix} & & \rho_{i,j+\frac{1}{2}} & & \\ & & & & \\ \frac{1}{2}\rho_{i-\frac{1}{2},j} & & -\Sigma & & \frac{1}{2}\rho_{i+\frac{1}{2},j} \\ & \frac{1}{2}\rho_{i-\frac{1}{4},j-\frac{1}{4}} & \rho_{i,j-\frac{1}{4}} & \frac{1}{2}\rho_{i+\frac{1}{4},j-\frac{1}{4}} & \end{bmatrix} \psi_{ij} = \frac{3h^2}{4}\eta(ih, jh).$$

Here, as before, the notation Σ represents the sum of the surrounding stencil entries.

2. Discretization of the Weldpool Problem on a Composite Grid

With an understanding of how FVE can be used to discretize a PDE on a composite grid, now consider discretization of the weldpool problem on a composite grid. Equations II.5 - II.8 are considered now on a composite grid. Figure 22 shows a two-level single patch composite grid, with grid spacing on the fine patch taken as half the grid spacing of the coarse grid.

The composite grid points that will be examined are those along the coarse-fine interface. Six grid points are labeled in Figure 22. These grid points are interior interface points (P and Q), Neumann boundary interface points (M and N) and Dirichlet boundary interface points (D and F). For the Dirichlet boundary, recall that since the Dirichlet conditions are

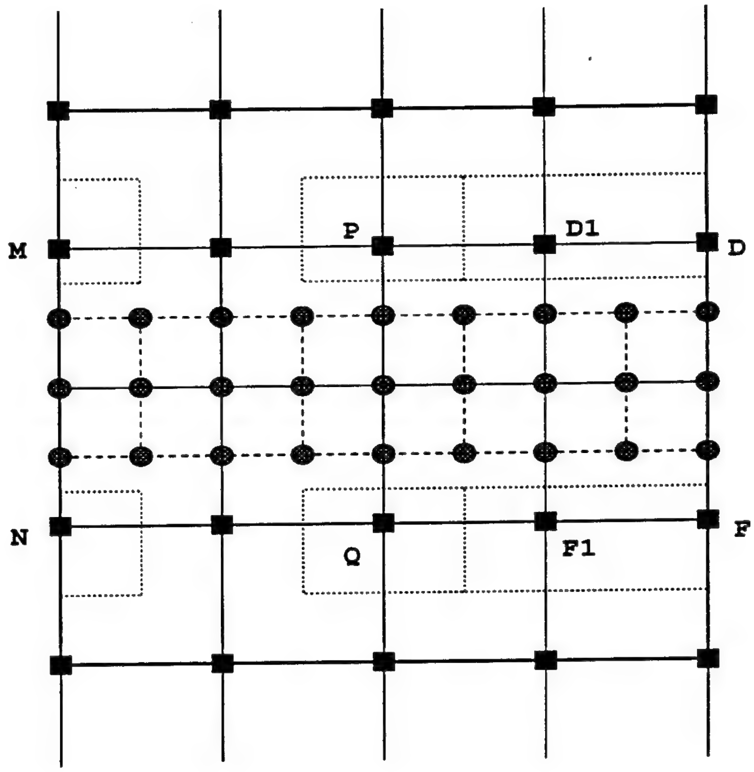

Figure 22. *A simple composite grid with volumes around interior boundary points P and Q, Neumann boundary points M and N, and Dirichlet boundary points D and F (extended volumes around D1 and F1).*

imposed directly on T, the control volumes are extended so that the control volumes normally associated with the boundary points are taken into account by an extended volume from the neighboring points (D1 and F1).

Beginning with the conduction equation for the solid,

$$\frac{\partial T}{\partial t} = Ma^{-1}\nabla^2 T, \tag{IV.25}$$

the discretization of the conduction problem resulted in the following integrals:

$$\begin{aligned} \sum_{i,j=0}^{m} \left[\frac{1}{g}\int_V \phi_{i,j}^{n+1} dV \right. &- \left. \alpha Ma^{-1}\int_S \nabla\phi_{i,j}^{n+1}\cdot \hat{n} dS\right] T_{i,j}^{n+1} \\ &= \sum_{i,j=0}^{m}\left[\frac{1}{g}\int_V \phi_{i,j}^{n} dV + (1-\alpha)Ma^{-1}\int_S \nabla\phi_{i,j}^{n}\cdot\hat{n}dS\right] T_{i,j}^{n}. \end{aligned} \tag{IV.26}$$

As before, the integrals over the basis functions are calculated by representing the basis functions as collections of triangular planes. The triangular planes are assembled to form the "hat" function. An interior grid point along the interface has seven triangular planes joined in this manner (Figure 21). A Neumann boundary point has four such triangular planes, as can be seen in Figure 23. A neighboring Dirichlet boundary point, with its extended volume, also has seven triangular planes. (Notice that the edge labeled "$SE3$" does not lie under the "hat" function of the neighboring Dirichlet boundary point $D1$.)

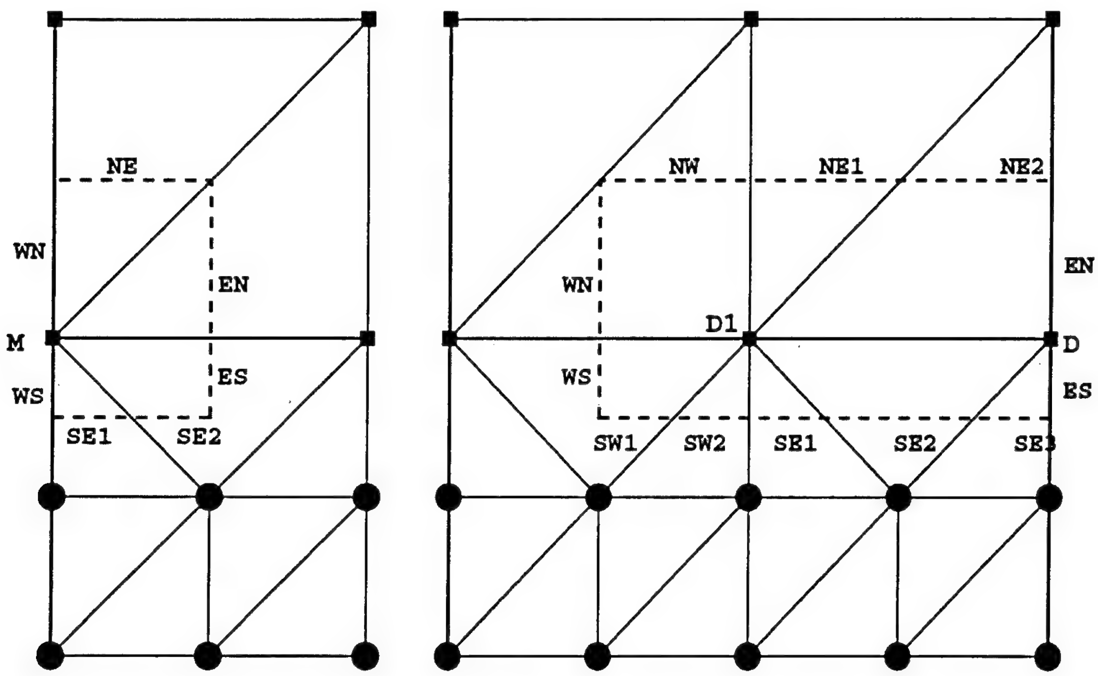

Figure 23. *Neumann boundary point (M) volume and Dirichlet boundary point (D) extended volume on a composite grid.*

Following the techniques outlined above for a uniform grid, interface point stencils have been computed for the volume and surface integrals of equation (IV.26). Referring to Figure 22, the stencils have been computed for interior interface grid points (P and Q) on the top and bottom boundary, and also for the Neumann (M and N) and neighboring Dirichlet

(D1 and F1) grid interface points on the top and bottom boundary. Below are the stencils for the volume and surface integrals for point P:

$$\int_{V_{i,j}} \left(\sum_{k,l} T_{k,l}\phi_{k,l} \right) dV = \frac{h^2 r}{384} \left[\begin{array}{ccccc} & & 32-5\epsilon & & 16+5\epsilon \\ 32-11\epsilon & & 192+5\epsilon & & 16+6\epsilon \\ & 32-10\epsilon & 48+4\epsilon & 16+6\epsilon & \end{array} \right] T_{i,j}$$

and

$$\int_{S_{i,j}} \left(\sum_{k,l} T_{k,l}\nabla\phi_{k,l} \cdot \hat{n} \right) dS = \frac{r}{32} \left[\begin{array}{ccccc} & & 32 & & \\ 16-9\epsilon & & -128 & & 16+9\epsilon \\ & 16-6\epsilon & 32 & 16+6\epsilon & \end{array} \right] T_{i,j}.$$

In the above stencils, $\epsilon \equiv \frac{h}{r}$ (r is the radial coordinate for the central point of the volume), and, as before, the 2π resulting from the integration in the θ direction has been factored out. The stencils for the remaining boundary points can be found in the Appendix.

The next stencils to be calculated are those belonging to the convection-diffusion equation. Again, observe that the discretization of the convection-diffusion equation,

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = Ma^{-1}\nabla^2 T, \tag{IV.27}$$

has the same form as the conduction equation for the solid except for the term

$$\int_V \nabla \cdot (\mathbf{u}T) dV. \tag{IV.28}$$

Interface point stencils have been computed for the integral in equation (IV.28). Referring to Figure 22, the stencils have been computed for interior interface grid points (P and Q) on the top and bottom boundary, and also for the Neumann (M and N) and neighboring Dirichlet (D1 and F1) grid interface points on the top and bottom boundary. The stencil

for point P has the following form:

$$\int_V \nabla \cdot (\mathbf{u}T)dV \quad = \quad \frac{1}{16}\begin{pmatrix} & & A & & B \\ C & & D & & E \\ & F & G & H & \\ & & & & \end{pmatrix} T_{i,j},$$

where the entries of the stencil are themselves stencils applied to the function Ψ. The entries are given here:

$$A = \begin{pmatrix} & & 2 \\ -4 & 2 & \\ & & \end{pmatrix} \Psi_{i,j} \qquad B = \begin{pmatrix} & -2 & \\ & & 2 \\ & & \end{pmatrix} \Psi_{i,j}$$

$$C = \begin{pmatrix} & 4 & \\ & -1 & \\ -3 & & \\ & & \end{pmatrix} \Psi_{i,j} \qquad D = \begin{pmatrix} & & -2 & & \\ 1 & & & & 1 \\ & 1 & & -1 & \\ & & & & \end{pmatrix} \Psi_{i,j}$$

$$E = \begin{pmatrix} & & -2 \\ -1 & & \\ & 3 & \\ & & \end{pmatrix} \Psi_{i,j} \qquad F = \begin{pmatrix} & & \\ 3 & -1 & \\ & -2 & \\ & & \end{pmatrix} \Psi_{i,j}$$

$$G = \begin{pmatrix} & & & \\ & 1 & & -1 \\ 2 & 2 & -4 & \\ & & & \end{pmatrix} \Psi_{i,j} \qquad H = \begin{pmatrix} & & \\ & & -2 \\ & 2 & \\ & & \end{pmatrix} \Psi_{i,j}$$

The stencils that have been developed are for the conduction equation in the solid (IV.25) and the convection-diffusion equation in the liquid (IV.27). Also, only stencils for upper interior interface grid points are displayed here. Many stencils have been developed and are displayed in the Appendix.

3. Development of the Weldpool Problem

The stencils developed in the previous section are for a model of the weldpool problem. The model used to develop the above stencils assumed a stationary boundary and considered only a flat interface between the two states (Figure 24).

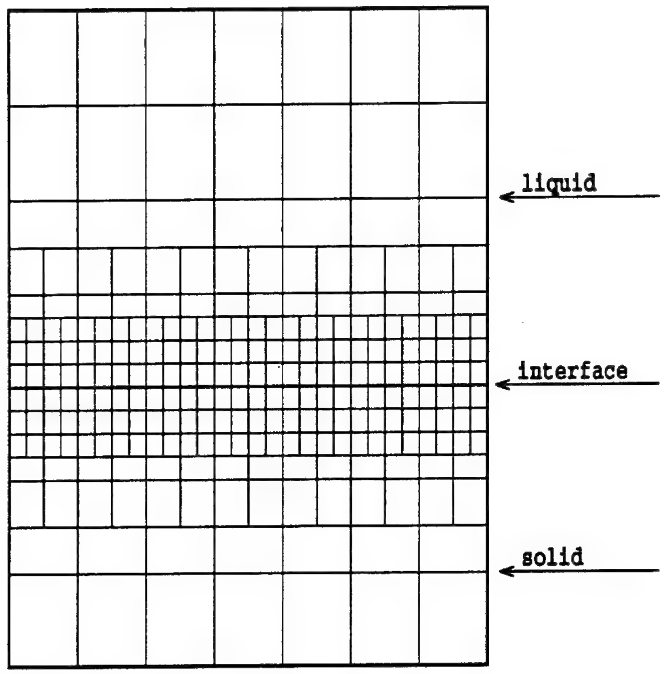

Figure 24. *An example of a composite grid for the simplified weldpool problem with a stationary, flat boundary. The grid spacing is finer along the solid-liquid interface.*

The real problem is far more complicated and requires some method for tracking the solid-liquid interface. The next step in solving the weldpool problem is to form the boundary into its naturally occuring shape and place a fine grid over the solid-liquid interface, where the boundary is still considered to be stationary. An example of this composite grid is found in Figure 25.

The next step is to consider the movement of the interface. The movement results from the melting and solidifying of the solid. This movement is governed by the local heat

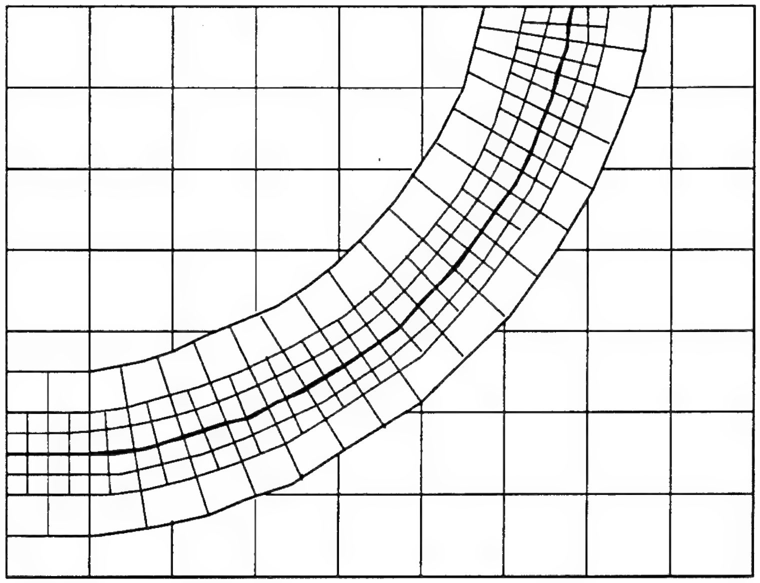

Figure 25. *An example of a composite grid for the weldpool problem with a stationary, curved boundary. The grid spacing is finer along the solid-liquid interface.*

balance around each interface point. A requirement for a control volume along this interface is that both the current time and the next time step must be contained in the volume. Not only the current position of the interface, but an approximation of the interface after one time step are required to construct the current local grid [Ref. 1]. This will require having an adaptive component in the time stepping. That is, at each step, the next position of the interface must be computed. If the change in position is too large and the new interface point lies outside the control volume, then the time step must be made smaller. A new local grid must also be computed at each time step. In this way, the control volumes are chosen in a manner that allows the interface to be moved.

These, and other complicating factors, indicate that a good deal of work remains before the full weldpool problem can be considered "solved".

V. CONCLUSION

The research for which this work is a part is designed to develop efficient numerical methods for solving the full weldpool problem. This work was undertaken with the goals of developing the FVE method for spatial discretization and finite differences for time differentiation to discretize the equations that describe the weldpool problem, and develop stencil equations for use in FAC. This task began with the application of finite differences to the time derivatives in the conduction problem, resulting in a time-stepping scheme that makes use of a weighted average at the current and the next time steps. FVE is then applied to the potential flow problem to develop the method for discretizing the spatial derivatives. Having developed these methods on model problems, the weldpool problem is tackled by applying finite differences and FAC to discretize the governing equations.

Multigrid is presented, including a development of the coarse grid, intergrid transfers and coarse grid correction. Multigrid is introduced here because of its effectiveness as an iterative method. Multigrid proves sufficient for solving problems on a uniform grid. However, the weldpool problem requires an extremely fine level of accuracy along the liquid-solid interface. To achieve this level of accuracy on a global grid would be too costly. Thus, FAC is developed to make use of composite grids. These grids allow finer grid spacing along the boundary of the molten pool and the solid, while still providing for coarser grid spacing where fine accuracy is not required. To arrive at FAC, several ideas are drawn from simpler, less efficient methods related to multigrid, including local multigrid and the multilevel composite grid method. A simple model of the weldpool problem is then discretized on a

composite grid using FVE. Stencil equations are developed for the conduction equation in the solid and the convection-diffusion equation in the liquid.

Finite Volume Element methods are shown to be an effective method for discretizing equations on a composite grid. The stencil equations that result from the discretization will be coded into FAC at a later date to determine their effectiveness. Future work will center on applying the FVE discretization and FAC to a moving, time-dependent liquid-solid interface.

APPENDIX. STENCIL EQUATIONS

Included here are the remaining stencil equations that were not included in the main body of the text.

The first of the stencil equations are for the conduction problem. The discretization of the conduction problem resulted in the following integrals:

$$\sum_{i,j=0}^{m}\left[\frac{1}{g}\int_V \phi_{i,j}^{n+1}dV - \alpha Ma^{-1}\int_S \nabla\phi_{i,j}^{n+1}\cdot\hat{n}dS\right]T_{i,j}^{n+1} \tag{.1}$$
$$= \sum_{i,j=0}^{m}\left[\frac{1}{g}\int_V \phi_{i,j}^{n}dV + (1-\alpha)Ma^{-1}\int_S \nabla\phi_{i,j}^{n}\cdot\hat{n}dS\right]T_{i,j}^{n}.$$

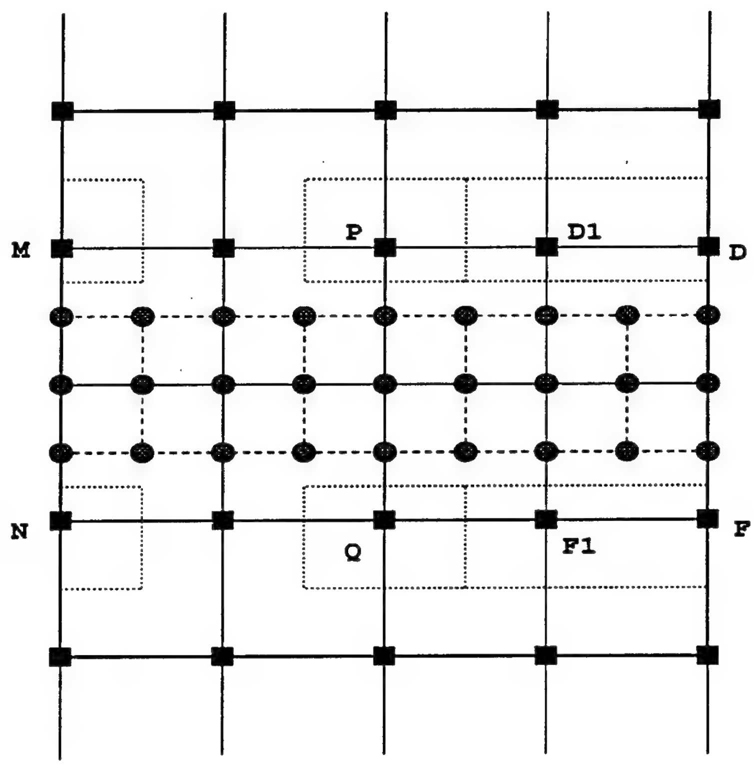

Figure 26. *A composite grid with volumes around interior boundary points P and Q, Neumann boundary points M and N, and Dirichlet boundary points D and F (extended volumes around D1 and F1).*

Referring to Figure 26, the stencils have been computed for the conduction problem above (.1) at interior interface grid points (P and Q) on the top and bottom boundary, and also for the Neumann (M and N) and neighboring Dirichlet (D1 and F1) grid interface points on the top and bottom boundary. Given here are the stencils for the volume and surface integrals for points Q, D1, F1, M, and N.

At point Q:

$$\int_{V_{i,j}} \left(\sum_{k,l} T_{k,l}\phi_{k,l} \right) dV = \frac{h^2 r}{384} \begin{bmatrix} & 16-6\epsilon & 48-4\epsilon & 32+10\epsilon & \\ 16-6\epsilon & & 192-5\epsilon & & 32+11\epsilon \\ & & & & \\ 16-5\epsilon & & 32+5\epsilon & & \end{bmatrix} T_{i,j}.$$

At point D1:

$$\int_{V_{i,j}} \left(\sum_{k,l} T_{k,l}\phi_{k,l} \right) dV = \frac{h^2 r}{384} \begin{bmatrix} & & 32-5\epsilon & \\ & & & \\ 32-11\epsilon & & 224+26\epsilon & \\ & 32-10\epsilon & 48+4\epsilon & 48+30\epsilon \\ & & & \end{bmatrix} T_{i,j}.$$

At point F1:

$$\int_{V_{i,j}} \left(\sum_{k,l} T_{k,l}\phi_{k,l} \right) dV = \frac{h^2 r}{384} \begin{bmatrix} & 16-6\epsilon & 48-4\epsilon & 48+20\epsilon \\ 16-6\epsilon & & 208+5\epsilon & \\ & & & \\ 16-5\epsilon & & 48+49\epsilon & \end{bmatrix} T_{i,j}.$$

At point M:

$$\int_{V_{i,j}} \left(\sum_{k,l} T_{k,l}\phi_{k,l} \right) dV = \frac{h^2 r}{384} \begin{bmatrix} & 8+\epsilon & & 16+5\epsilon \\ & & & \\ & 104+24\epsilon & & 16+6\epsilon \\ & 32+6\epsilon & 16+6\epsilon & \\ & & & \end{bmatrix} T_{i,j}.$$

At point N:

$$\int_{V_{i,j}} \left(\sum_{k,l} T_{k,l}\phi_{k,l} \right) dV = \frac{h^2 r}{384} \begin{bmatrix} & 16+2\epsilon & 32+10\epsilon & \\ & 88+19\epsilon & & 32+11\epsilon \\ & & & \\ & 24+6\epsilon & & \end{bmatrix} T_{i,j}.$$

At point Q:

$$\int_{S_{i,j}} \left(\sum_{k,l} T_{k,l} \nabla \phi_{k,l} \cdot \hat{n} \right) dS = \frac{r}{32} \left[\begin{array}{ccccc} & 16-6\epsilon & 32 & 16+6\epsilon & \\ 16-9\epsilon & & -128 & & 16+9\epsilon \\ & & 32 & & \end{array} \right] T_{i,j}.$$

At point D1 (Notice here that $T_{i,j}$ is known for the furthest right entries of this stencil. These three stencil elements can be pulled from the stencil.):

$$\int_{S_{i,j}} \left(\sum_{k,l} T_{k,l} \nabla \phi_{k,l} \cdot \hat{n} \right) dS = \frac{r}{32} \left[\begin{array}{ccccc} & & 32 & & 16+12\epsilon \\ 16-9\epsilon & & -128-9\epsilon & & -32-18\epsilon \\ & 16-6\epsilon & 32 & 16 & 32+30\epsilon \end{array} \right] T_{i,j}.$$

At point F1 (Notice here that $T_{i,j}$ is known for the furthest right entries of this stencil. These three stencil elements can be pulled from the stencil.):

$$\int_{S_{i,j}} \left(\sum_{k,l} T_{k,l} \nabla \phi_{k,l} \cdot \hat{n} \right) dS = \frac{r}{32} \left[\begin{array}{ccccc} & 16-6\epsilon & 32 & 16 & 32+30\epsilon \\ 16-9\epsilon & & -112+7\epsilon & & -16-10\epsilon \\ & & 16-6\epsilon & & 4\epsilon \end{array} \right] T_{i,j}.$$

At point M:

$$\int_{S_{i,j}} \left(\sum_{k,l} T_{k,l} \nabla \phi_{k,l} \cdot \hat{n} \right) dS = \frac{r}{32} \left[\begin{array}{ccc} 16+4\epsilon & & \\ -64-21\epsilon & & 16+9\epsilon \\ 16+2\epsilon & 16+6\epsilon & \end{array} \right] T_{i,j}.$$

At point N:

$$\int_{S_{i,j}} \left(\sum_{k,l} T_{k,l} \nabla \phi_{k,l} \cdot \hat{n} \right) dS = \frac{r}{32} \left[\begin{array}{ccc} 16+2\epsilon & 16+6\epsilon & \\ -64-21\epsilon & & 16+9\epsilon \\ 16+4\epsilon & & \end{array} \right] T_{i,j}.$$

In the above stencils, $\epsilon \equiv \frac{h}{r}$ (r is the radial coordinate for the central point of the volume) and, as before, the 2π resulting from the integration in the θ direction has been factored out.

Discretization of the convection-diffusion equation,

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}\mathrm{T}) = \mathrm{Ma}^{-1}\nabla^2\mathrm{T}, \tag{.2}$$

has the same form as the conduction equation for the solid except for the term

$$\int_V \nabla \cdot (\mathbf{u}\mathrm{T})\mathrm{dV}. \tag{.3}$$

Interface point stencils have been computed for the integral in equation (.3). Again referring to Figure 26, the stencils have been computed for interior interface grid points (P and Q) on the top and bottom boundary, and also for the Neumann (M and N) and neighboring Dirichlet (D1 and F1) grid interface points on the top and bottom boundary. The stencils given here ard for points Q, D1, F1, M, and N.

At point Q, the stencil has the following form:

$$\int_V \nabla \cdot (\mathbf{u}\mathrm{T})\mathrm{dV} \;=\; \frac{1}{16}\left(\begin{array}{ccccc} & A & B & C & \\ D & & E & & F \\ G & & H & & \end{array}\right) T_{i,j},$$

where the entries of the stencil matrix are themselves stencils, applied to the function Ψ. The stencil entries are given as:

$$A = \left(\begin{array}{ccc} & 2 & \\ -2 & & \\ & & \end{array}\right)\Psi_{i,j} \qquad B = \left(\begin{array}{cccc} & -4 & 2 & 2 \\ -1 & & 1 & \\ & & & \end{array}\right)\Psi_{i,j}$$

$$C = \left(\begin{array}{ccc} & -2 & \\ & -1 & 3 \\ & & \end{array}\right)\Psi_{i,j} \qquad D = \left(\begin{array}{ccc} & 3 & \\ & & -1 \\ -2 & & \end{array}\right)\Psi_{i,j}$$

$$E = \begin{pmatrix} & -1 & & 1 & \\ 1 & & & & 1 \\ & & & & \\ & & -2 & & \end{pmatrix} \Psi_{i,j} \qquad F = \begin{pmatrix} & & & & -3 \\ & & -1 & & \\ & & & & \\ & & 4 & & \end{pmatrix} \Psi_{i,j}$$

$$G = \begin{pmatrix} & & \\ 2 & & \\ & & \\ & & -2 \end{pmatrix} \Psi_{i,j} \qquad H = \begin{pmatrix} & & \\ & 2 & -4 \\ & & \\ 2 & & \end{pmatrix} \Psi_{i,j}$$

At point D1, the stencil has the following form:

$$\int_V \nabla \cdot (\mathbf{u}\mathrm{T})\mathrm{dV} \quad = \quad \frac{1}{16} \begin{pmatrix} & & A & \\ & & & \\ B & & C & \\ & D & E & F \end{pmatrix} T_{i,j},$$

where the entries of the stencil matrix are themselves stencils, applied to the function Ψ.

The stencil entries are given as:

$$A = \begin{pmatrix} & & 2 \\ & & \\ -4 & 2 & \\ & & \end{pmatrix} \Psi_{i,j} \qquad B = \begin{pmatrix} & & 4 \\ & & \\ & & -1 \\ & -3 & \end{pmatrix} \Psi_{i,j}$$

$$C = \begin{pmatrix} & & -2 & & 4 \\ & & & & \\ 1 & & & & \\ & 1 & & -4 & \end{pmatrix} \Psi_{i,j} \qquad D = \begin{pmatrix} & & \\ 3 & & -1 \\ & & -2 \end{pmatrix} \Psi_{i,j}$$

$$E = \begin{pmatrix} & & & \\ & & & \\ & 1 & & -1 \\ 2 & 2 & -4 & \end{pmatrix} \Psi_{i,j} \qquad F = \begin{pmatrix} & & \\ & & \\ 3 & & -3 \\ & 2 & -2 \end{pmatrix} \Psi_{i,j}$$

At point F1, the stencil has the following form:

$$\int_V \nabla \cdot (\mathbf{u}\mathrm{T})\mathrm{dV} \quad = \quad \frac{1}{16} \begin{pmatrix} & A & B & C \\ D & & E & \\ & & & \\ F & & G & \end{pmatrix} T_{i,j},$$

where the entries of the stencil matrix are themselves stencils, applied to the function Ψ.

The stencil entries are given as:

$$A = \begin{pmatrix} & & & & \\ & 2 & & & \\ -2 & & & & \\ & & & & \\ & & & & \end{pmatrix} \Psi_{i,j} \qquad B = \begin{pmatrix} & & & & \\ & -4 & 2 & 2 & \\ -1 & & 1 & & \\ & & & & \\ & & & & \end{pmatrix} \Psi_{i,j}$$

$$C = \begin{pmatrix} & & & & \\ & & -2 & 2 & \\ & & -3 & & 3 \\ & & & & \\ & & & & \end{pmatrix} \Psi_{i,j} \qquad D = \begin{pmatrix} & & & & \\ & 3 & & & \\ & & -1 & & \\ & & & & \\ -2 & & & & \end{pmatrix} \Psi_{i,j}$$

$$E = \begin{pmatrix} & & & & \\ & -1 & & 4 & \\ 1 & & & & \\ & & & & \\ & & -4 & & \end{pmatrix} \Psi_{i,j} \qquad F = \begin{pmatrix} & & & & \\ & & & & \\ 2 & & & & \\ & & & & \\ & & -2 & & \end{pmatrix} \Psi_{i,j}$$

$$G = \begin{pmatrix} & & & & \\ & & & & \\ & 4 & & -4 & \\ & & & & \\ 2 & & & -2 & \end{pmatrix} \Psi_{i,j}$$

At point M, the stencil has the following form:

$$\int_V \nabla \cdot (\mathbf{u}\mathrm{T})\mathrm{dV} \quad = \quad \frac{1}{16} \begin{pmatrix} & & A & & B \\ & & & & \\ & & C & & D \\ & & E & F & \\ & & & & \end{pmatrix} T_{i,j},$$

where the entries of the stencil matrix are themselves stencils, applied to the function Ψ.

The stencil entries are given as:

$$A = \begin{pmatrix} & & & -2 & 2 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{pmatrix} \Psi_{i,j} \qquad B = \begin{pmatrix} & & & -2 & 4 \\ & & & & \\ & & & & -2 \\ & & & & \\ & & & & \end{pmatrix} \Psi_{i,j}$$

$$C = \begin{pmatrix} & & -4 & & \\ & & & & \\ & & & & 1 \\ & & 4 & -1 & \\ & & & & \end{pmatrix} \Psi_{i,j} \qquad D = \begin{pmatrix} & & & & 2 \\ & & & & \\ & & 1 & & \\ & & & -3 & \\ & & & & \end{pmatrix} \Psi_{i,j}$$

$$E = \begin{pmatrix} & & & & \\ & & & & \\ & & -1 & & 1 \\ & & -4 & 4 & \\ & & & & \end{pmatrix} \Psi_{i,j} \qquad F = \begin{pmatrix} & & & & \\ & & & & \\ & & & & 2 \\ & & & -2 & \\ & & & & \end{pmatrix} \Psi_{i,j}$$

At point N, the stencil has the following form:

$$\int_V \nabla \cdot (\mathbf{u}\mathrm{T})\mathrm{dV} \;=\; \frac{1}{16} \begin{pmatrix} & A & B & \\ & C & & D \\ & & & \\ & E & & \end{pmatrix} T_{i,j},$$

where the entries of the stencil matrix are themselves stencils, applied to the function Ψ.

The stencil entries are given as:

$$A = \begin{pmatrix} & & & & \\ & & -2 & 2 & \\ & & & & \\ & & & & \\ & & & & \end{pmatrix} \Psi_{i,j} \qquad B = \begin{pmatrix} & & & & \\ & & -2 & & \\ & & -1 & & 3 \\ & & & & \\ & & & & \end{pmatrix} \Psi_{i,j}$$

$$C = \begin{pmatrix} & & & & \\ & & -4 & 1 & \\ & & & & 1 \\ & & & & \\ & & 2 & & \end{pmatrix} \Psi_{i,j} \qquad D = \begin{pmatrix} & & & & \\ & & & -3 & \\ & & -3 & & 2 \\ & & & & \\ & & 4 & & \end{pmatrix} \Psi_{i,j}$$

$$E = \begin{pmatrix} & & & & \\ & & & & \\ & & 2 & & -4 \\ & & & & \\ & & 2 & & \end{pmatrix} \Psi_{i,j}$$

The stencils that have been included here are for the conduction equation in the solid (IV.25) and the convection-diffusion equation in the liquid (IV.27). The stencils here are for

the interface, either in the interior or along the boundary. Stencil equations for the vorticity and stream functions, equations (II.7) and (II.8), still remains for future computation.

REFERENCES

[1]David Canright and Van Emden Henson. Towards an FVE-FAC approach to determining thermocapillary effects on weld pool shape. working draft, unpublished 1995.

[2]William F. Ames. *Numerical Methods for Partial Differential Equations.* Computer Science and Scientific Computing. Academic Press, San Diego, third edition, 1992.

[3]G. D. Smith. *Numerical Solutions of Partial Differential Equations: Finite Difference Methods.* Oxford Applied Mathematics and Computing Science Series. Oxford University Press, Oxford, Great Britain, third edition, 1985.

[4]Stephen F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations*, volume 6 of *Frontiers in Applied Mathematics.* Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[5]William L. Briggs. *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, 1987.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5101	2
3. Professor Van Emden Henson, Code MA/Hv Department of Mathematics Naval Postgraduate School Monterey, CA 93943-5216	2
4. Professor David Canright, Code MA/Ca Department of Mathematics Naval Postgraduate School Monterey, CA 93943-5216	2
5. Professor Richard Franke, Code MA/Fe Department of Mathematics Naval Postgraduate School Monterey, CA 93943-5216	1
6. LT Thomas E. Rogers P.O. Box 3793 Pensacola, FL 32516	1